Appendix S

Software Source Selection

Content

S1.0 Tab 1: Source Selection Under Acquisition Reform

Under acquisition reform, the offeror's process and past performance are considered as significant criteria. Therefore you, as software acquisition managers and software engineers participating in source selection, must evaluate the contractor's processes and experience to select the best offeror capable of providing a quality system with the lowest development and life cycle risks. The newly revised DoD Directive 5000.1, Defense Acquisition, 15 March 1996, describes "broad management principles that are applicable to all DoD acquisition programs." It states the following about acquiring software-intensive systems:

Software is a key element in DoD systems. It is critical that software developers have a successful past performance record, experience in the software domain or product-line, a mature software development process, and evidence of use and adequate training in software methodologies, tools, and environments.

Regardless of acquisition size, you should evaluate these areas during source selection. Remember, while it is necessary for a contractor to have a mature software development process, you should examine the process that particular division or component within the organization proposes to use on your program. The parent organization as a whole might have a mature process; however, certain divisions or components within the organization might not be as experienced in or knowledgeable of that process. To assess this, you should ask if the division or component you are evaluating has successfully adopted the parent organization's process. You should also examine the division's or component's specific experience in using these processes in your application domain.

One last consideration is source selection for post-deployment software support. If you perform a source selection at this time, you must make sure the proposed post-deployment software support process will provide at least the same level of quality software as the development process provided. To do this, you should use the same rigorous methods for source selection of the post-deployment software support organizations that you used for the development organization.

S2.0 Tab 2: Software Capability Evaluation (SCE)

As mentioned in Volume 1, Chapter 10, Software Development Maturity, there are two software development capability assessment methods effective for determining the maturity of an organization's software development (and support) process — the Software Development Capability Evaluation (SDCE), developed by Aeronautical Systems Center, and the Software Capability Evaluation (SCE) developed by the Software Engineering Institute. While this section provides information on how to implement the SCE, if you are acquiring a C3 or ground electronics system, your are encouraged to contact Electronic Systems Center (ESC) for assistance in conducting the SCE.

While this section does not address the SDCE, you can find out more information (including about training) by contacting Aeronautical Systems Center (ASC) . For additional assistance with MIS acquisition, please contact the Standard Systems Group (SSG). For Air Force in-house software development organizations with questions on Software Process Improvement and Software Maturity Assessments, contact the Air Force C4 Agency.

S2.1 SCE Implementation Guidelines

Software Capability Evaluation (SCE) offers a means to evaluate an organization's software process capability, that is, how well an organization manages the process it uses to create software. SCE provides a way to compare a development organization's software process against a predefined standard. The purpose of these guidelines is to standardize the application of SCE on source selections. The Software Engineering Institute's (SEI) Capability Maturity Model[SM] (CMM[SM]) is the basis for this SCE appraisal. The SCE appraisal is intended to be considered as an integral part of the source selection evaluation, however, the SCE evaluation team may operate independent of other area/factor source selection evaluators. SCE results should be evaluated consistent with evaluation criteria specified in the Request For Proposal (RFP).

S2.1.1 Applicability

SCE applies to all source selections for Management Information Systems (MIS) and Command, Control, Communications, Computer, and Intelligence (C4I) Systems with software development costs greater than $10 million. Software development includes: development of new code, modification of existing code, and integration of software modules. Source selections with software development costs less than $10 million should consider the use of SCEs based on a cost/benefit tradeoff and the goal of acquisition streamlining. An SCE should always be performed on prime contractor Offerors. The only exception is when the proposed prime does not do, and never has done, software development, and is acting as only a "general contractor." In this exception, the proposed prime must not impose any process guidance on proposed subcontractors that affect the sub's software development activities.

Conducting multiple SCEs on an Offeror's team is encouraged if one or more proposed subcontractors are to perform significant software development. All proposed subcontractors performing more than $10 million or 35% of the software development using their own processes should be evaluated. SCEs are normally not applicable for source selections for the acquisition of commercial-off-the-shelf (COTS) or non-developmental item (NDI) software. SCEs should be performed when NDI software modifications to satisfy Air Force requirements or "glue code" development to link COTS packages will cost more than $10 million.

S2.1.2 SCE Appraisal

The source selection related SCE appraisal should investigate/cover, as a minimum, all Key Process Areas (KPA) and goals for the Repeatable and Defined maturity levels described in the CMM.SM The appraisal should not tailor the KPAs or goals specified in the CMM.SM The SCE appraisal team may be independent but is part of the source selection team. The SCE appraisal should be conducted as soon as possible after a competitive range decision is made and discussions with contractors are authorized. Normally, each SCE includes a 4-5 day site visit, in addition to preparation and wrap-up time for each site visit for a total of up to ten (10) days.

The SCE appraisal should be conducted by a team trained in the CMMSM version to be used and comprised of members from the Air Force and/or an Air Force approved independent organization. If a contractor is used to conduct the SCE appraisal, FAR Subpart 9.5, Organizational and Consultant Conflicts of Interest, shall be adhered to. In selecting programs for review, the priority should be programs currently being worked, or recently completed, by the Offeror rather than programs most similar to the acquisition. The programs must be from the same organization. The reviewed programs should be approximately the same size, from the same development site, and from the same broad domain. Offeror must be put on notice in the Request For Proposal (RFP) that process documentation from selected programs must be available at the unclassified level.

An Offeror's software engineering/development practices should be considered validated if: (1) a written and approved procedure for a practice exists; (2) the procedure implementation is effective for the organization; (3) evidence exists showing that procedures are followed; (4) evidence exists that training for the procedure is planned, funded, scheduled, required, and accomplished in a timely manner; and (5) the procedure has been institutionalized. An institutionalized practice or procedure is one that has been in place and practiced for greater than 12 months. Practices and procedures less than six months old may be considered a process improvement activity. Practices and procedures in place and practiced for less than 12 months but longer than 6 months may, by SCE team consensus, be considered institutionalized.

There are three components of the CMM[SM] reference model that can be rated: goals, KPAs, and maturity level. SCE results are documented as KPA findings of general observations, strengths, weaknesses, and process improvement activities. A strength is a particular part of the software process capability that is sufficiently robust to mitigate the development risk due to software process. A weakness is a particular part of the software process that has characteristics that increase the risk due to software process. A process improvement activity is a practice or procedure that is not yet institutionalized and indicates potential mitigation of risk due to software process. Maturity level ratings are optional since the rating itself provides minimal visibility into the state of an appraised contractor's software process. All findings are determined by team consensus.

- **Goal**. A goal is satisfied when the associated findings indicate that the goal is implemented, as defined in the CMM,[SM] with no significant weaknesses or that an adequate alternative exists, and is institutionalized, as defined in paragraph 4.5 above.
- **KPA**. A KPA is satisfied when all goals for that KPA have been investigated/covered and rated as satisfied. A KPA is weak if one or more goals for it are not satisfied. A KPA is assigned "Not Rated" if any of the goals for the KPA are not investigated/covered.
- **Maturity Level**. A maturity level is achieved when all KPAs for that level and all of the levels below it have been investigated/covered and rated as satisfied.

At the conclusion of each SCE, a findings exit brief should be provided at the site. The exit brief is to provide a courtesy one way information flow of the draft findings to the Offeror before the SCE team leaves the site, and allow the Offeror to provide information and/or artifacts that may have been overlooked. The exit brief should include the PCO name, address, contact method, and instructions for the Offeror to respond or comment on the draft findings presented by the SCE team. The RFP should explain the exit briefing rules.

S2.1.2.1 Source Selection Evaluation of SCE Results

Normally, the SCE will be designated as a factor in the management area. The source selection evaluation team should evaluate the SCE results using the following procedures.

The SCE shall be a significant factor. Do not include in the SCE factor any augmentation elements. Any augmentation elements should be evaluated in a separate factor.

In addition to affecting the SCE factor rating, the SCE results may affect proposal risk assessments relevant to other factors.
Evaluation of SCE subfactors should be standardized. When converting the SCE subfactors into a factor color, the following criteria should be used:

- **BLUE**. There are no weak KPAs at the Repeatable and Defined maturity levels.
- **GREEN**. There are no weak KPAs at the Repeatable maturity level *and* four (4) or less weak KPAs at the Defined maturity level.
- **YELLOW**. No KPAs at the Repeatable maturity level are rated weak *and* five (5) to seven (7) KPAs at the Defined maturity level are rated weak; or
  - One (1) KPA at the Repeatable level is rated weak *and* six (6) or less KPAs the Defined level are rated weak; or
  - Two (2) KPAs at the Repeatable level are rated weak *and* five (5) or less KPAs at the Defined level are rated weak; or
  - Three (3) KPAs at the Repeatable level are rated weak *and* four (4) or less KPAs at the Defined level are rated weak.
- **RED**. There are four (4) or more weak KPAs at the Repeatable maturity level or eight (8) or more weak KPAs in the Repeatable and Defined maturity levels.
  - When multiple SCEs for an Offeror — proposed subcontractor team are conducted, the following standards should be used to determine the team's SCE factor color rating and proposal risk.
  - Where the proposed prime's color rating is lower than one or more of its proposed subcontractors, the team's color rating should always reflect the rating of the proposed prime.
  - When the team members each use their own processes, the lowest color rating among them should determine the proposal's color rating.
  - If a higher color rated proposed prime imposes its higher level processes on the proposed subcontractor(s), the higher color rating may be used, with proposal risk being assessed to indicate that one or more lower rated proposed subcontractors have never used the process.
  - Where the team members are equally color rated and they will integrate their processes, the color rating should be the color rating of the individual team members and the proposal risk should be other than LOW.

S2.1.2.2 Post Contract Award

Contracts should be structured to allow the performance of one or more SCEs subsequent to contract award to assure the evaluated level is maintained and/or to verify progress against Software Process Improvement Plans (SPIPs).

S2.2 Evaluating ADA Experience During SCE

Objective. This provides an outline of the issues that should be addressed when assessing a contractor's ability to develop programs in Ada during a Software Capability Evaluation (SCE).

Background. A SCE can provide a snapshot of a contractor's past process implementation, current process activities, and futures process potential. SCE's are based on the Software Engineering Institute (SEI) Capability Maturity Model[SM] (CMM[SM]) assessments. The SEI CMM[SM] Version 1.1 is a good starting point for assessing the capability of a contractor. This can provide the basis for a similar assessment of a language capability. Once the language is known (be it Ada, Fortran, or C), we can ask the contractor a new set of questions focused on that language. Most of the issues are not specific to any particular language, e.g., Ada. The approach to take is to identify the issues needed to access the contractor's capabilities in a given language and then to fill in the details when the language is Ada.

Proposal. There are six SEI CMM[SM] Key Process Areas (KPAs) that can be tailored to include assessment of the contractor's Ada capability. The first two KPA's, Software Tracking and Oversight, Software Quality Assurance (SQA), can be used to get a snapshot of how a contractor performs on Ada Programs. The following are those KPA's and some issues that can be addressed during the evaluation.

KPA — Training Program. The SCE team could first learn about the contractor's Ada capabilities from reviewing their training program. The training plans should show what type Ada training is planned. For example, is there Ada training for other than programmers (e.g., program managers, SQA, and SCM personnel). The employee's training records should reveal who has been trained and was the training taken at the appropriate time (i.e., prior to the start of working on an Ada program).

KPA — Software Program Planning. This KPA allows the SCE team to see how a contractor develops estimates for schedules, manpower, facilities, and sizing (lines-of-code). When reviewing the contractor's estimating methodologies and procedures, the team would look for an Ada influence. The contractor's management should show an understanding of the Ada language and its use. The people selected for programs should have used the language within the domain of their program. Domain experience is more important than language experience. If the language used in previous programs is similar, then this should be an advantage. While reviewing the contractor's program plans, the team can see if the selected hardware meets the need of the developers. Does the contractor plan for adequate file server processing capacity and disk storage to support the program? Ada tends to need more computer cycles and disk storage than other languages. Finally, the management must demonstrate a commitment to doing what is necessary to make the program a success (e.g., additional training, software tools, hardware, etc.).

KPA — Software Program Tracking and Oversight. Once the team has seen how the contractor plans for Ada programs, then they should see how the plans work on those programs. The tracking metrics should be tailored for the use of Ada. The data collected from these metrics should be reported to management. The management should show a commitment to taking any corrective actions necessary based on the results from the metrics.

KPA — Organization Process Definition. This KPA's force is on the company standards and procedures. The SCE team would be looking for Ada programming standards and procedures. These standards should be up to date and easily available to all programmers. The Unit development folders should show signs that the standards are being used.

KPA — Software Product Engineering. Under the Software Product Engineering KPA, the SCE team should be looking at how the software environment is set up for building the Ada software. For example, are the following tools used?

- CASE tools,
- Configuration management tools,
- Compilers,
- Integration tools, and
- Code generators that support Ada development.

These tools should be integrated with the contractor's overall software development methodology and software development process. It is also important for the contractor to have experience with these tools.

KPA — Software Quality Assurance (SQA). The final KPA that can be used by the SCE team in assessing a contractor's Ada capability is the SQA KPA. Under this KPA, the SCE team should review the SQA procedures to see if any are covering the company Ada standards. The SQA training records should be reviewed for some type of Ada training.

Summary. The above KPAs and related issues are just a starting point for a SCE team to use in assessing a contractor's Ada capabilities. Additional information may be learned under the other KPAs not listed here. To better help a SCE team in their review, the following sample questions related to the above listed KPAs has been developed.

S2.2.1 SCE Key Process Areas (KPAs) for Ada Evaluation

The following lists some of the Key Process Areas (KPAs) from the Software Engineering Institute (SEI) Capability Maturity Model[SM] (CMM[SM]) and questions for a Software Capability Evaluation team to use in assessing a contractor's Ada capabilities. In addition are some non-SEI CMM[SM] KPAs and questions that should also be considered.

S2.2.1.1 KPA Questions

Training program. Does the training planning include Ada training and is the training provided? Is there Ada training for other than programmers (e.g., program manager, SQA, and SCM personnel)? Is there any on-the-job training? Are experienced programmers assigned to work with the under experienced programmers? Is follow-up training provided? When are the people trained? Have they taken all required training prior to being assigned to a program using Ada? Are they encouraged and do they take additional or follow-up training?

Software program. Does the contractor's estimating methodologies and procedures for schedules, manpower, and sizing have an Ada influence?

Planning. Are the people planned for the program those who have used the language within the domain of the program? Have they used the proposed tools? How well does the management understand the language and its use? Is the management committed to doing what is necessary to make the program a success? Does the detailed software development process support the contractor's management techniques? Is there adequate hardware available to meet the needs of the developers? Does each developer have a workstation? Is there adequate file server processing capacity and disk storage to support the team?

Software program tracking and oversight. Are the tracking metrics tailored for the use of Ada? Does management review the metrics and are corrective actions taken?

Organization process definition. Are there company standards and procedures for Ada? Are they tailored for each Ada program? Are they used by the programmers? Are they reviewed on a regular basis and updated as needed? Do the Unit Development Folders show signs the standards are being used?

Software product engineering. Are the following tools used?

- CASE tools,
- Configuration management tools,
- Compilers,
- Integration tools, and
- Code generators that support Ada development.

How are these tools integrated with the contractor's overall software development methodology and software development process? What experience does the contractor have with these tools?

Software quality assurance. Review SQA procedures for any covering the company Ada standards. The SQA training records should be reviewed for some type of Ada training.

S2.2.1.2 Non-KPA Questions

Reuse. Do they have a reuse component in their process? Does it support the language being used (e.g., Ada)? Do they have and use a corporate reuse library? How is reuse coupled back to the development process? How are reusable components tested and validated?

COTS. Do they have experience in integrating COTS products in general and with products they are using on this program? Do they have experience integrating COTS products written in other languages with the program's language (e.g., Ada)?

S2.3 Subprocess Area Selection Tables

The tables in this appendix are provided as an aid to help SCE teams select critical subprocess areas during Step 5. The tables were created by the SCE program members at the SEI for guidance only. SCE teams are expected to use their experience and judgement to select critical subprocess areas based on the requirements of the particular development. Factors considered in selecting critical subprocess areas are the following:

- What processes would an organization need to manage the aspects of the program which are new to the organization?
- If the product being developed is new to the end user, what processes will the development organization need to manage the anticipated requirements changes?
- What are the basic processes that a development organization would need for any software development effort?

S2.3.1 How to Read the Tables in This Section

This appendix contains a table for each key process area (KPA) in the *Repeatable* and *Defined* levels. The tables contain the following columns.

Subprocess areas column. Each row under this column corresponds to a subprocess area associated with the KPA. Some of the subprocess areas contain other subprocess areas. These "*higher-level*" subprocess areas are indicated by boldface type.[1]

Major attributes columns (ApD, Pt, Ps, Tw, and Sub). An "*X*" in the column for an attribute indicates that the subprocess area listed in that row may be important to the development organization for managing the risk associated with a lack of experience relative to that attribute. These columns correspond to the five major attributes from the Experience Table created in Step 4. The Experience Table shows where any of the development organizations may lack experience with regard to some attribute of the new program.

Operational precedence (Op) column. An "*X*" in this column indicates that the subprocess area listed in that row may be important for managing the level of requirements changes which may be anticipated if end users do not have experience with similar products. The Op column corresponds to the *operational precedence* attribute from the Target Product Profile developed by the sponsor. This attribute indicates the degree to which the product being developed may be new to the end user.

Nucleus capability (*) column. An "*X*" in this column indicates that the subprocess area listed in that row is part of the recommended *nucleus capability*. Nucleus capability refers to a basic set of subprocesses which are needed for almost any software development.

S2.3.2 Repeatable Level Key Process Areas (KPAs)

Key to Abbreviations:

ApD	Application Domain
Tw	Type of Work
Op	Operational Precedence
Pt	Product Type
Sub	Subcontracting
*	Nucleus Capability
Ps	Product Size

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
General Management Functions							
Committed management process	X	X	X			X	X
Compliance to organizational standards							
Taking corrective action; issue/action item tracking			X				X
Review and oversight: oversight by senior management and management reviews							X
Tracking; actual vs. estimate comparison; commitment evidenced by reviews of compliance							X
Customer interface	X					X	
Usage and collection of performance data							X

Repeatable Level Key Process Area: Program Management

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Integrated Software Management							
Risk management; recognition of risk events; cost, software technology, resources, and schedule	X	X	X			X	
Tailoring and selection of project process and its support environment							
Maintenance of process performance database							
Coordination between project groups	X	X	X			X	

Repeatable Level Key Process Area: Program Management (cont.)

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Requirements Management	X	X				X	
Requirements allocation							
Requirements change							X
Requirements implication evaluation			X				
Matching software architecture to requirements; transforming requirements into top-level design							

Repeatable Level Key Process Area: Program Management (cont.)

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Subcontracting							
Subcontractor selection							
Contracting: subcontract process							
Coordination of work with subcontractor							
Subcontractor monitoring							

Repeatable Level Key Process Area: Program Management (cont.)

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Testing							
Preparing to carry out testing; test procedures							
Carrying out test operations							
Reviewing test scenarios, testbeds, and test cases							
Regression testing							X

Repeatable Level Key Process Area: Program Management (cont.)

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Project Planning							
Size estimation: software development resources, costs and critical target and host computer resources; the scope of work and effort has a basis in reality	X	X	X			X	X
Cost estimation; cost has documented correspondence to estimate size and schedule; software responsibility, software engineering technical direction	X	X	X			X	X
Planning: resource planning and management for project's software size, cost and schedule, software development plan, the software life cycle model, planning schedules, software schedules							X
Commitment process during change	X	X	X			X	X
Project manager's participation with the project proposal team	X	X	X			X	
Usage of software process database							
Integration of technical direction, engineering tools and methods into planning process, engineering and technical reviews of plans	X	X	X			X	
Product capacity tracking, critical target computer resources							

Repeatable Level Key Process Area: Program Planning

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Configuration Management							
Status report, monitoring, configuration responsibility							X
Change control process, standard forms for reporting errors			X				X
SCM plan; baselining of software engineering products and process specifications; a configuration management repository for the software baselines; software baseline audits	X	X					X
Release of software baseline products							
Library support system			X				
Configuration control board							

Repeatable Level Key Process Area: Configuration Management

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Software Quality Assurance							
Auditing: SQA objective evidence of audits							X
Noncompliance resolution	X	X	X			X	X
Reporting chain: SQA group reports, independent authority							X
SQA plan							X
SQA concurrence on milestone progress	X	X					
SQA group participation							
Oversight for all process support systems; e.g., corrective action system; data collection of defects; earned value of system deviation handling	X	X					

Repeatable Level Key Process Area: Software Quality Assurance

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Software Engineering Process Group							
Assignment of full-time resources, establishing and supporting							X
Coordination of review with senior project technical staff, analysis, and evaluation of software process definition, responsibility assignment	X	X	X			X	X
Planning systems and software process improvement; review of existing and proposed process standards							
Defining training requirements	X	X				X	

Defined Level Key Process Area: Software Engineering Process Group

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Standards and Procedures							
Planning standard software process development	X	X	X			X	
Implementing standard software process development							
Process assets; a process library system; library of software process specifications; software process database maintenance; tailoring the organization's standard software process	X	X	X			X	
Standards for software development folders							X
Review standards							
Human-machine interface standards							

Defined Level Key Process Area: Standards and Procedures

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Software Product Engineering							
Integrating the project's process with the software architecture: process change and technology transition review	X	X				X	X
Investigating software engineering tools and methods; tool selection and use with gathering of performance data							
Developing and maintaining the project's software architecture							
Reviewing the system/software testing							
New technologies		X				X	

Defined Level Key Process Area: Software Product Engineering

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Training							
Planning/procuring training courses for training curriculum, courses							
Job analysis to support each project's training needs							
Communicating and keeping track of delivered training; schedules for all professional and technical staff; records of training							
Delivering training; management support							
The organization's training program; training requirements							

Defined Level Key Process Area: Training

SUBPROCESS AREAS	MAJOR ATTRIBUTES						
	ApD	Pt	Ps	Tw	Sub	Op	*
Peer Reviews							
Planning/assisting peer reviews; technical review	X	X	X			X	X
Schedule, process for technical reviews							
Conducting peer reviews							X
Review assignments							
Peer review performance; organizational database of review activities; cost; peer review result handling							

Defined Level Key Process Area: Peer Reviews

S2.4 Notes

1. Most of these became KPAs in the Capability Maturity ModelSM (CMM[SM]) Version 1.1 [Paulk 93a], and were established in anticipation of that version of the CMM[SM]. Some of the subprocess areas distinguished in this manner are at the wrong maturity level relative to CMM[SM] Version 1.1; however, this does not affect how an SCE is conducted, because maturity level scores are not calculated. It does alter the category the findings are reported under, because findings are consolidated by KPA.
2. The abbreviation Ps stands for "*Product Size*." Product Size refers to the "*Size*" attribute.

S2.5 SCE Text for Inclusion in Instructions to Offerors

S2.5.1 Section L

The following sample text illustrates how SCEs might be inserted within Section L or M of the RFP. These examples assume the SCE will be used as a specific criterion for source selection.

S2.5.2 Sample 1

Software Engineering Capability. The Government will evaluate the software process by reviewing the offeror's Software Process Improvement Plan and by using the Software Engineering Institute (SEI) developed technique, the Software Capability Evaluation. The Government will determine the software process capability by investigating the offeror's current strengths and weaknesses in key process areas defined in the SEI report CMU/SEI-TR-11 "*Characterizing the Software Process: A Maturity Framework.*" The Government will perform an SCE of each offeror by reviewing current programs at the site proposed on this contract. The evaluation will be an organizational composite. It will be substantiated through individual interviews and reviews of documentation, of the offeror's strengths and weaknesses in key process areas relative to maturity level three; i.e., the extent to which an offeror meets or exceeds maturity level three criteria. The on-site evaluators may be separate and distinct from the proposal evaluation team and may include a government contracting representative. The evaluators will have been trained and experienced in conducting SCEs.

S2.6 SCE Text for Inclusion in Instructions for Preparation of Proposals (IFPP)

NOTE: Instructions for Preparation of Proposals provide guidance to offerors as to how they should prepare their proposal. The following text requests the offeror to provide program profiles, organization charts, sample documentation, and a software process improvement plan. It also requests the offeror to provide the SCE team with facilities during the site visit.

The technical proposal shall include the offeror's response to the software evaluation process. The offeror shall provide the following information to assist the Government's preparation for the Software Capability Evaluation of each offeror:

1. The offeror shall complete the Program Profile form for 7-9 major software engineering development programs. All programs should be drawn from the same site and organization (e.g., profit center) bidding on this solicitation. One of these programs must include the (proposed) software development effort and the others should be programs that are near completion or completed within the last three years. These programs should be as similar as possible in scope and magnitude to the (proposed) effort. The programs should be from programs where the offeror was the prime contractor, at least one program should include a development where another subcontractor developed portions of the software, and as least one program should be an Ada program, more if applicable. Program Profiles from Special Access Programs are discouraged. For offerors with fewer than 7 programs at the bidding site, submit information for as many programs as are available.
2. Section C, Tab 1, contains the questionnaire outline and report form that should be used to generate the evaluation profiles for each of the programs. Respond to the SEI questions with a Yes or No answer. For each "yes" response, please note the mechanism or document for justifying the response on a separate form.
3. The offeror shall provide program-level and higher-level organization charts. The organization charts should contain individual's names and job titles and indicate how the programs above are related to each other. If there are departments that the software programs rely on, these too should be positioned on the organization chart (e.g., training, Software Engineering Process Group , quality assurance, configuration management, standards, policy and procedures).
4. The offeror shall provide a draft Software Development Plan (SDP) and a Software Standards and Procedure Manual (SSPM). If there are "generic" SDPs and SSPMs those are preferred; otherwise, select a sample SDP and SSPM from the program that has the most representative SDP.
5. The offeror shall submit their site's Software Process Improvement Plan, in the form of their choosing, with their proposal. The document shall be no longer than 15 pages. The Software Process Improvement Plan shall be detailed enough for the offeror to communicate their current software process capability, specific planned improvements, dedicated resources, effort estimates, and a time phasing of those improvements to bring the offeror's software process maturity to the organization's desired maturity level.
6. After the proposal is received, the Government will coordinate a site visit with the offeror to discuss the questionnaire responses and conduct the Software Capability Evaluation (SCE) at the offeror's location. The offeror shall provide a point of contact and phone number for the coordination of all SCE activities. So that the site visit will go smoothly, the Government will list details about the site visit during the coordination process; e.g., interview schedules, documentation requests, facilities for the evaluation team. The offeror shall be notified approximately two working days prior to the site visit of the programs to be examined. The site visit dates selected by the Government are not open for discussion.
7. During the site visit, the SCE team will need a secure meeting room capable of accommodating at least eight people. The offeror shall have a copy of the organization's software standards, procedures and/or operating instructions, and organizational charts for the programs being reviewed in the meeting room when the SCE team arrives. All interviews conducted as part of the SCE shall be done in private, one individual at a time. The SCE team may be separate and distinct from the proposal evaluation team.
8. If security authorization is necessary for the members of the evaluation team, a Fax number and telephone number of the contractor's security office should be provided along with a list of any other pertinent information required to obtain security approval.

S3.0 Tab 3: Sample RFP Preparation Checklists

S3.1 Sample Questionnaire for Site Visit Preparation

The following questions are examples of what you should consider as you develop your site visit checklist. [SOURCE: Yourdon, Edward, Decline and Fall of the American Programmer, Yourdon Press, Englewood Cliffs, New Jersey, 1992].

- Does this company care about software quality? Does it care enough, for example, to delay putting a new system into production because its software reliability models indicate an unacceptable number of latent errors? Does it have software reliability models?
- Does this company care about its people? Has it invested time and money to train its software development managers to do a better job in hiring people? Does it invest an adequate amount of time training its technicians, or does it assume that its software engineers are replaceable commodities? Does it use modern "performance management" methods to ensure that its corporate goals are aligned with personal consequences of those goals? Do the people in the organization understand what the organizational goals are, and how they are supposed to fit into those goals?
- Does this company use modern programming tools, languages and methodologies, as opposed to assembly language and the waterfall life cycle.
- Does this company measure everything it does in the software arena? Does it measure the process of software development as well as rate the final product? Does it have a separate software metrics group? Are size, effort, schedule, defects, and rework measured routinely? Are the metrics used in a positive way, so that everyone in the organization can see how they improve?
- Does this company support the concept of software reusability? More important, does it provide some incentive (for example, cash royalties) to its software engineers to create reusable components? Has it considered a separate "Software Parts Department" whose only job is to create reusable components? Does it estimate the degree of expected reusability at the beginning of programs and base its schedules and resource requirements on that estimate?
- Does this company have CASE tools? Does it believe that CASE tools are like toothbrushes, that is, they're not meant to be shared? Does it provide an adequately equipped PC or workstation for everyone?

S3.2 Program Profile Outline

The following outline is a sample program profile that can be referenced in the RFP. Six to nine of these forms, for different programs, should be filled out by each contractor.

- **Program Name**: (name of program listed on the contract)
- **Program Number**: (unique identifying number on the contract)
- **Program Type**: (e.g., scientific, human-machine, business, control, support software)
- **Customer**: (the agency that procured the software and a point of contact within that agency)
- **Subcontractors/Prime Contractors**: (list any subcontractors employed on the program or list the prime contractor if the offeror was a subcontractor)

- **Current Phase**: (identify the current phase of the software development process; e.g., requirements definition, detailed design, code & unit test, integration test, maintenance)
- **Location**: (primary site of the software development effort)
- **Start Date**: (starting date of the contract)
- **Design Completion Date**: (estimated or actual)
- **Code Completion Date**: (estimated or actual)
- **End Date**: (contract completion date)
- **Team Size**: (peak man-month period and average man-years over the contract period)
- **Estimated KSLOC and Function Points**: (estimated/actual thousand source-lines-of-code (KSLOC)) and function points.
- **Programming Languages**: (percentage of KSLOC in languages (e.g., Ada, FORTRAN, Pascal, C, Assembly))
- **Target Hardware System**: (computer on which software executes)
- **Development Hardware System**: (host computer for the compiler and support environment)
- **Applicable Standards**: (e.g., MIL-STD-498)
- **Cost**: (actual/estimated dollars spent to date/completion)
- **SEI Questionnaire**: (the attached questionnaire and its answer sheet should be completed for each of the programs)
- **Organization Chart**: (Most recent organization chart for each program with titles and individual names. This chart should identify the individual responsible for the following activities: program management, system engineering, software program management, software engineering, software quality assurance, software configuration management, subcontractor control, simulation, integration and testing and other technical software activities.)

S3.3 Proposal Evaluation Checklist

The following checklist is provided as an aid for software development proposal evaluation.

S3.3.1 Program Management

- What was the software manager's involvement with the proposal?
- How is software progress tracked? Management reviews? Frequency?
- Who will approve software schedules? Cost estimates?
- How are issues raised, tracked, and conflicts resolved?
- What will be the software manager's reporting chain?
- How does the software requirements team relate to the software design team?
- How much manager visibility into integration and test will be necessary?
- What will be the relationship between the System Engineer and Software? How will tradeoffs be made?
- Is senior management briefed regularly on software status?

S3.3.2 Subcontractor Management

- What is the subcontractors' development process?
- How will qualified software subcontractors be selected?
- Do the subcontractor's standards, procedures, process comply with the prime contractors'?
- How should the results and performance to commitments be tracked?
- Is the subcontract manager knowledgeable of and trained in the software?
- Are there periodic technical reviews & interchanges with subcontractor?
- Does the prime's Software Quality Assessment and Configuration Management monitor sub's SQA & CM?
- Do the prime's senior management review the status of the subcontractor regularly?

S3.3.3 Metrics Management

- Is design progress, test progress and staffing measured?
- Is integration progress measured?
- Is software size overtime and memory utilization measured?
- Is throughput and I/O channel utilization measured?
- Is progress tracked and reported to the PM regularly?
- Are technical, schedule, cost, and resources plans prepared?
- How are software size, cost and schedules established? How are document procedures established?
- Document Commitments: Who commits, size, cost and schedules?
- Are there policy exits for resource planning and commitments?
- Are the software managers trained on software estimation?
- Are actual versus planned estimates recorded and compared?
- Is there a central estimation manager and data base for accuracy?

S2.3.4 Software Quality Assurance Management

- Is there an independent reporting chain?
- Are audits conducted at all phases of life cycle and line activities?
- How is it ensured that audits are representative?
- Does SQA have adequate resources?
- Does SQA audit subcontractors?
- Are deviations handled according to documented procedures?
- Does senior management review SQA activities regularly?
- Is SQA authority and concurrence required?

S3.3.5 Configuration Management

- How can requirements, design, and code changes be controlled?
- How can interface changes be controlled?
- Is there traceability for requirements, design and code?
- Is there a tool to help control versions and builds?
- Are parameters established for regression testing?
- Are baselines established for tools, change log, and modules?
- Does the CM plan include staff, schedule, response, resources, tools, and facilities
- Does the library system store work products and prevent unauthorized change?
- Does the document change request process include check in/out, review and regular testing?
- Is there a document Change Control Board and a change proposal process?
- Is there a change log that tracks open/closed change requests?

S3.3.6 Peer Reviews Management

- Are design, code, and test case peer reviews conducted?
- Who and how many people attend?
- Are documented procedures and checklists used?
- Are the peer reviews included in the Software Development Plan and are they published?
- Are statistics compiled on the type, severity, and location of errors?
- Are statistics compiled on the time to prepare, review, and correct elected errors?
- How are errors tracked to closure?
- Does SQA audit peer review activities?

S3.3.7 Training

- How are CM and Quality Assurance leaders trained?
- Are moderators and developers included in peer reviews?
- Do program managers participate in software estimation and peer reviews?
- Do software supervisors participate in QA, CM, estimation and peer reviews?
- Do software developers participate in peer reviews, software development process and tools?
- Do training resources include money, facilities, tools and schedules?
- Is there a corporate training policy supported by a training manual?
- Are program training needs identified and planned?
- Are job functions mapped to training?
- Do training records include people and courses?

S3.3.8 Standards Management

- Do standards include coding, unit development folders, and man-machine interface standards?
- Do standards include generic SDP, a QA plan and a CM plan?
- How are standards enforced?
- How and when are standards updated?
- What is the assigned response for updating standards and policy?

S4.0 Tab 4: Sample Paragraphs for RFP Inclusion

S4.1 Software Quality Requirement

Software quality requirements will be specified for the program. The development of these requirements shall be the responsibility of the program office. The program office will work together with the end-user of the system to generate requirements based on an analysis of the system requirements, life expectancy, development costs and user concerns. Example user concerns to consider are performance (e.g. reliability, usability and efficiency), design architecture (e.g. maintainability and correctness) and re-engineering (e.g. reusability, interoperability and portability). Software quality requirements will be specified and documented within the baselined Software Requirements Specification (SRS). A hierarchical quality model of quality factors, criteria and metrics will be used to predict software quality. Factors representing the user's concerns will be decomposed (using relevant standards and guidebooks) into software oriented characteristics. Measures of these characteristics (i.e. metrics) will also be defined. The specified model will apply to all software development phases and products. Quality progress will be reported and reviewed at each major program milestone. All open and closed software quality problems will be tracked and reported. The achievement of software quality requirements will be demonstrated, using industry accepted measures of operational quality (e.g. reliability = mean-time-to-failure), during integration testing. Failures will be categorized according to an Government approved some severity standard.

S4.2 Software Testing Requirement

In addition to functional testing of the software to assure compliance with requirements, the software will be tested such that 100% of the software branches (i.e., decision to decision statements) are exercised prior to release in the field. Reasons for not achieving 100% execution coverage must be formally documented in the Software Test Report.

Software tools (i.e., test coverage analyzers) to automate the branch testing process are available. Intrusive analyzers insert software code into the software under development to capture and record the execution coverage and are appropriate for non-real-time software developments. If a software product under development must operate in real-time, if it is highly memory constrained, or if the software units are very large, non-intrusive analyzers should be used. Non-intrusive analyzers use a separate hardware processor to capture and record this same execution coverage information.

S4.3 Software Life Cycle Development and Support Environment Requirement

An automated computer-based software life cycle development and support environment will be used by the contractor. Development of the environment's requirements shall be the responsibility of the program office. The environment should provide the following capabilities: 1) specification of the life cycle software development process and the monitoring/enforcement of that process, 2) integration of Computer-aided Software Engineering (CASE) and other tools supporting the various interphase activities of the life cycle, and 3) interphase support including program management, configuration management and baselining, document/specification generation, traceability and change impact analysis.

S4.4 Software Life Cycle Development Technology Scalability Requirement

An automated, computer-based software life cycle development and support environment will be used by the contractor. Development of the environment's requirements shall be the responsibility of the program office. The ability of the environment's hardware/software complex (including each of its associated CASE tools) to adequately and efficiently support the breadth of software under development (i.e., scalability to the size of the problem) will be a primary consideration.
Reusable Software Requirement

As part of the SDP, reuse software engineering and planning shall be addressed. The SDP shall contain a WBS that includes the establishment and implementation of a reuse program. Reuse shall be an integral part of software development planning, review, audit and reporting. As part of the contractor's SEE, a Software Reuse Library shall be established and maintained after appropriate review and approval by the Government.

S5.0 Tab 5: Source Selection for Software Supportability

S5.1 Instructions to Offerors (Section L)

In addition to specifying proposal form and content, the Instructions to Offerors should require submission of a Software Development Plan and Software Quality Program Plan as part of the proposal. The SDP will include the offeror's software development and management concepts, procedures, and metrics for controlling and assessing progress during the development process.

S5.2 Supportability Issues

The following supportability issues must be covered in the Instructions to Offerors:

- The methodology used to perform software sizing and cost estimating and the approach to be followed during software development
- The rationale used for computer resource timing and sizing estimates and description of how spare I/O utilization (channels or data rates), CPU throughput utilization, memory utilization requirements will be met;
- A description of any teaming and subcontractor arrangements;
- The skill levels required for computer resources development and their availability within the corporate structure;
- The method to be used for risk control;
- Any planned use of firmware;
- Any plans for reusing or modifying existing software;
- A clear definition of all assumptions used during proposal preparation;
- Plans for the development of prototype software;
- Plans and procedures for generating and using software metrics.
- A disclosure statement of defect removal efficiency. This should include their definition of defects and what defects are included in the metric and the method of calculating the metric.

S5.3 Additional MIL-STD-498 Considerations

- The offeror should address the manner in which they will comply with their Requirements for Software Standards, how this will be achieved and how such compliance will be measured. The offeror should describe proposed software development methodologies to be incorporated in any resultant contract.
- The offeror should document the manner in which compliance with Category and Priority Classifications for Problem Reporting, will be achieved and describe the problem reporting system to be used in any resultant contract.

- The offeror's proposal to the items above should be part of the technical volume of the proposal and not be required as part of the contract.
- The offeror should document the manner in which compliance with Evaluation Criteria will be achieved in the software development effort if the offeror is awarded the contract. These include: internal consistency; understandability; traceability to indicated documents; consistency with indicated documents; appropriateness of analysis, design, and coding techniques used; appropriateness of allocation of sizing and timing resources; adequacy of test coverage of requirements.

NOTE: The offeror may propose, subject to government approval, additional criteria or alternate definitions for any of the criteria.

- The offeror shall also provide examples of software documentation (e.g., software specifications, source code listings, software test reports) prepared on other software development efforts (the Government's source selection team can then evaluate the supportability of the proposed documentation.) The offeror should describe the process, techniques, methodologies, and metrics to be used and define acceptable (i.e., pass) criteria (minimum, range, or maximum) for each proposed evaluation criteria test environment (STE) (including tools therein) (see definitions below) proposed to develop software for the system. The offeror should also describe the environment proposed to be delivered to (assuming the contract requires such delivery) or to be used by the Government to support the system's software. *[The offeror should be required to submit metrics on this issue to help government evaluators determine the quality of the environment proposed for delivery to the Government.]*
- Offeror should document the factory software engineering environment (SEE), including tools therein. The plans should address how the offeror will evolve the factory environment into the supporting environment. This should not include the concept of developing a separate support environment. The evolution should include the constant updating and refining of the factory environment to meet all needs of the supporters and then be transitioned to the supporting/maintaining organization.
- The plans should also describe how the offeror will install the support environment at the supporting/ maintaining organization, load the environment with all program software/data and hardware (e.g., operational software/data, all development/ test tools, hardware configurations, master engineering data repository, and administrative practices to be used for software support) and use the environment as the only source of information/tools to support the initial operational test and evaluation (IOT&E), as well as initial block changes to the system (while under interim contractor support).
- The plans should describe any differences in tools between the factory environment and that envisioned for the software support activity and plans to ensure that tools differences will not adversely impact the supportability of the software.

NOTE: If too much documentation is required for submission to the Government, it may exceed page count restrictions.

- The offeror should document the approach to be used in evaluating the quality of software and software development processes; i.e., how the offeror will comply with proposed evaluation criteria during the period of the contract. In addition, the offeror shall identify, explain (with rationale), and provide pass (as in pass/fail) criteria for each process and product metric used. This document shall contain a step-by-step sequence of quality-related activities to include the data collection process, scoring algorithms, reporting, and corrective action.

S5.4 Additional AFSCP/AFLCP 800-45, *Software Risk Abatement,* Considerations

The offeror should document the approach to be used in managing risk in developing software and integrating it in the system. The offeror should be required to quantify performance, support, cost, and schedule risk factors (this should be part of the offeror's Software Development Plan).

S5.5 Sample Section L

The following information is useful for developing Instructions to Offerors (ITO) (as related to software supportability concerns):

1. Submit Volume XXX and completed questionnaires from A Method for Assessing the Software Engineering Capability two weeks prior to submission of Volumes XXX.

2. Volume I. TECHNICAL
Volume I shall describe the complete proposed Reliability and Maintainability Plan and engineering programs and shall not exceed xxx pages. Volume I shall be divided into two books, marked and placed in separate three-ring or spiral binders. Each book shall be arranged as described below.

2.2. Volume I, Book II. Engineering Program and Design

Book II shall detail the proposed engineering program. As a minimum, the following information shall be included.

2.2.1. Describe the overall engineering development and design program including major activities and an integrated schedule.

2.2.1.1. Identify the overall engineering development schedule and specific integration program activities such as design requirements analysis, testing, software development, support equipment development, and management processes for controlling the development effort.

2.2.1.2. Provide an overall technical description of the total program. Identify significant benefits of design features proposed including commonality considerations among subsystems (including support equipment, maintenance trainers, and aircrew training devices), between aircraft types, and between aircraft mission design series. Include, as a minimum:

2.2.1.2.1. Software design, development, and integration efforts for each subsystem. Include the top-level description of each computer software configuration item (CSCI), identify and justify the computer languages used, and estimate the size of each CSCI. Identify if the CSCI currently exists, will be modified, or will be developed.

2.2.1.2.2. Describe the overall built-in-test (BIT) approach for each subsystem and how it will test subsystem and subsystem-to-aircraft interfaces.

2.2.1.3. Define the draft subsystem specification and development plan for the major subsystems. The specifications shall be sufficiently detailed, as a minimum, to include descriptions of:

2.2.1.3.1. Growth potential of each LRU with respect to the number of circuit card assembly (CCA) expansion slots available and the type of functional enhancements (such as additional memory, processor, or input/output CCAs).

2.2.1.3.2. Significant components at the SRU level, such as embedded computers and memory devices. Identify and justify the intended processors to be used, estimated lines of code, throughput, memory, and growth capacity requirements.

2.2.1.3.3. Identify the significant benefits of design features proposed including commonality considerations among CCAs or SRUs.

2.2.1.4. Within the draft subsystem specification and development plan for the xxx subsystem, provide additional detailed descriptions of the following:

2.2.1.4.1. Identify commercial software to be used. Describe the level of documentation available, to be developed, and how the Government will support the commercial software.

2.2.1.4.2. Describe how the XXX subsystem software development and design approach will allow modification of display page formats or information, incorporate additional pages, provide for growth in number of display units and display avionics management units.

2.2.2. Describe relevant engineering development experience of the technical personnel proposed for this program. Include specific information on planned contribution to this program for each person identified. Be specific about team members with experience on at least two programs of similar scope, and where similar engineering tasks were accomplished.

2.2.4. Describe engineering development facilities (laboratories), staffing, and equipment planned for use on this program. Identify the respective availability of each resource and plans to acquire resources not currently available.

2.2.5. Define the preliminary support equipment (SE) program. Describe the overall program for designing, developing, and testing the proposed support equipment for the XXX system.

2.2.5.1. Describe how the support equipment selection and development processes integrate with the BIT software development effort and maintenance procedures development.

2.2.5.2. Describe test program sets (TPSs) to be used with both existing and newly developed test stands. Include a description of the TPS hardware and software requirements and identify compatible automated test equipment (ATE).

2.2.6. Describe the overall software development and management program.

2.2.6.1. Describe the software development approach, analysis methods, and integrated schedule for completing the software configuration items.

2.2.6.2. Define all software development tools that will be used including such applications as compilers, assemblers, debuggers, editors, linkers, loaders, and configuration management programs. Define the computer and operating systems on which each software tool will be used. Describe how these tools will be made available to the Government.

2.2.6.3. Describe relevant software development experience of the technical and management personnel proposed for this program. Include specific information on planned contribution to this program for each person identified. Be specific about certifications held by inspection personnel.

2.2.6.4. Describe software development facilities, staffing, and equipment planned for use on this program. Identify the respective availability of each resource and plans to acquire resources not currently available.

2.2.6.5. Describe how your software development program will support the independent validation and verification (IV&V) effort. Describe the data and documentation which shall be provided as part of the IV&V effort. Describe how IV&V personnel will be accommodated.

2.2.6.6. Each offeror may be visited by a government software assessment team (SWAT) as part of a site survey to assess software engineering capabilities. The survey will be conducted using A Method for Assessing the Software Engineering Capability, provided as an attachment to this RFP. Complete the questionnaire below to prepare for the SWAT survey.

2.2.6.6.1. Provide a completed software assessment program Form 01 (programs profile summary) and Form 02 (answers to the software assessment questionnaire) for six on-going software development programs (representative of all phases of software development) and the proposed (program name) software development efforts. The type of information required is indicated on the forms provided and shall be used to prepare responses (attachment following this section of the RFP). Provide the completed forms to the program contracting officer (PCO) separately from the proposal (address listed in paragraph XXX). The forms shall be delivered in accordance with the letter from the PCO coordinating the dates for the SWAT survey at each contractor's facility.

2.2.6.6.2. Each offeror will be notified by separate letter from the PCO to coordinate the SWAT survey visit. The team will conduct interviews with software program leaders, quality personnel, system integrators (software testing), and configuration management personnel to discuss the answers provided on the forms and assess software engineering capabilities. Additional documentation will be requested to validate responses to the questionnaires. Documentation may include, but not limited to, cost estimating worksheets, unit development folders, software development procedures, organizational charts, software quality audit reports, and software change requests.

2.2.7. Describe the overall test and evaluation (T&E) program.

2.2.7.1. Describe all computer models, test stands, and hot mock- ups needed to ensure accurate integration and interface requirements analysis and design verification. Include the basic concept of operation for each test stand and hot mock-up. Provide a description of your modeling tools for structural and stress analysis. Identify the availability of each resource and plans to acquire resources not currently available.

2.2.7.2 Describe the integration of software development and management activities with detailed test and evaluation activities.

2.2.7.3 Describe test and evaluation facilities, staffing, and equipment planned for use on this program. Describe capability to provide supply support and maintenance to the T&E level of flight testing. Identify the respective availability of each resource and plans to acquire resources not currently available.

3. REFERENCES

a. Department of Defense

(1) Directives/Instructions

(2) Standards

[The following standards should only be cited in accordance with DoDD 5000.1 and DoD 5000.2-R]

(a) MIL-STD-498, *Software Development and Documentation*

(b) DoD-STD-1467 (AR), *Software Support Environment*

(c) MIL-HDBK-347, *Mission-Critical Computer Resources Software Support*

(d) DoD-STD-1703, *NSA/CSS Software Product Standards Manua*l

(3) Other

(a) Defense Systems Management College (DSMC), *Mission Critical Computer Resources Management Guide*

b. Air Force

c. Other

[The following references provide additional guidance, and should come from industry first, then (if applicable) government sources. Again, refer to the 5000 series for guidance.]

(1) AFOTEC Pamphlet 99-102, Volume 3, *Software Supportability Evaluation Guide (Contact AFOTEC/SAS)*

(2) AFOTEC Pamphlet 99-102, Volume 5, *Software Support Resources Evaluation Guide,*

(3) AFSCP/AFLCP 800-45, *Acquisition Management Software Risk Abatement (Contact HQ AFMC/EN)*
(4) AFMCP 800-51, *Software Development Capability Assessment (Contact HQ AFMC/EN)*
(5) ASC Pamphlet 63-103, *Software Development Capability Capacity Review (Contact ASC/EN)*
(6) CMU/SEI-94-TR-06, *Software Capability Evaluation (SCE), Version 2.0, Method Description (Contact Software Engineering Institute at Pittsburgh PA)*
(7) RADC-TR-85-37, *Specification of Software Quality Attributes*, Volumes I-III,

4. DEFINITIONS

a. Software supportability: characteristics of software and computer support resources that affect the ability of software support activities to correct errors, add system capabilities, delete features, and modify software to be compatible with hardware changes. It should be noted that as the Air Force moves toward truly open systems, the need to modify software to be compatible with hardware changes should no longer exist.

(1) Organization: Software possesses the characteristic of organization when the documentation is logically partitioned into sets of volumes and document development conventions have been followed. It also measures how easily specific information is located within the documentation. Another factor is how well the documents have been divided along functional lines. A hierarchical partitioning of the system's documentation of less detail to descriptions of more detail should reflect the partitioning of software.
(2) Descriptiveness: Software documentation possesses the characteristic of descriptiveness when it contains information about its intent, assumptions, inputs, processing, outputs, components, and revision status. Documentation should have a descriptive format and contain useful explanations of the software program design.
(3) Traceability: Software documentation possesses the characteristic of traceability when information about all program elements, and their implementation, can be traced between all levels of lesser and greater detail (up and down in the system hierarchy). Program elements consist of, but are not limited to, data flow, control flow, algorithms, variables, and constants. Software may be well written and well described but still lack a clearly defined trail between top level requirements and detailed implementation. The software maintainer must be able to trace any particular element from higher levels of program description down to executable code, and from executable code to higher levels of program description. Traceability should also be evident from requirements through the design to the tests which verify the design.
(4) Modularity: Software possesses the characteristic of modularity when the software design is based on a logical partitioning/grouping of software and its parts/logically related abstractions and based on minimized module/unit interdependence. Software that is the easiest to understand and change is composed on independent modules. The fewer and simpler the connections between modules, the easier it is to understand each module without reference to other modules. Reducing connections between modules also minimizes the paths along which errors can propagate into other modules of the system. Modularity also implies that a module consists of only a few easily recognizable functions which are closely related with a minimal number of links to other modules.
(5) Consistency: Software possesses the characteristic of consistency when products correlate and contain uniform notation, termination, and symbology. The use of standards and conventions in documentation, flow chart construction and certain conventions in input/output processing, module interfacing, naming of modules/variables, etc., are typical indicators of consistency. This characteristic permits for the software maintainer to concentrate on understanding the true complexities of an algorithm, data structure, etc.

(6) Simplicity: Software possesses the characteristic of simplicity when it reflects the use of singularity concepts and fundamental structures in organization, language, and implementation techniques. The use of high order language as opposed to an assembly language makes a program relatively simpler to understand because there are fewer discriminations that have to be made. The number of operators, operands, nested control structures, nested data structures, executable statements, statement labels, decision parameters, etc., will determine to a great extent how simple or complex the source code is.

(7) Expandability: Software possesses the characteristic of expandability when a physical change to information, computational functions, data storage, or execution time can be easily accomplished once the nature of what is to be changed is understood. The design should allow for flexible timing, reasonable storage margin, parameterized constants, and indentured numbering scheme for source listings that easily accommodate changes.

(8) Testability: Software possesses the characteristic of testability when it contains aids which enhance testing. The documentation should describe how well the program has been designed to include test aids (instruments), while the source listings should illustrate how the code is implemented to allow for testing. The software should be designed and implemented so testability is either embedded within the program or can be easily inserted into the program or is available through a combination of these capabilities.

(a) Testability provides information on the logical build of functions or processes of the designed/developed software from the development phase of its individual computer software units (CSUs), into its integration phase of CSUs into computer software components (CSCs), and CSCs into computer software configuration items (CSCIs).

(b) Testability includes the testing of security software requirements for compartmented, and/or multilevel security modes of operation.

(c) Testability reflects the "*as-designed*" requirements of the software as they are developed into the "*as-built*" capabilities of the final software product.

(9) Convention: Software possesses the characteristic of convention when the software products correlate and contain uniform notation, terminology, and symbology. The use of standards in documentation, flow chart, or program design language construction and certain conventions in input/output processing, error processing, module interfacing, naming of modules/variables, etc., are typical reflections of convention.

(10) Design: Software possesses the characteristic of design when programs are formed using a structured method consisting of functional parts which are interrelated, yet independent of one another.

(b) NOTE: Software requirements traceability is inherent in software supportability; that is, all requirements should be traceable through the documentation to the appropriate test procedure and area of code for each specific requirement.

(11) Reusability: Software created during the development process that posses the potential for reuse within the same program or other programs.

b. Other Software Supportability Characteristics

(1) Portability: Software possesses the characteristic of portability when it is relatively easy to rehost software from one hardware platform to another hardware platform. This characteristic will require initial software development to consider future rewriting for adaptation to new hardware platforms.

(2) Machine Independence: Software possesses the characteristic of machine independence when it can be run on any hardware platform without needing to be modified to do so.

(3) External documentability: Software possesses the characteristic of external documentability when documentation (e.g., hierarchy charts, flow control charts, compilation sequence, data flow diagrams, and general explanations of what and how the software is used) matches the as-built software exactly.

(4) Coupling: Minimum degree of interaction between CSUs.
(5) Cohesion: Maximum degree of interaction within a CSU.
(6) Structured: One entry and exit per CSU.
(7) Standardization of Naming Conventions: Use of uniform notations for naming data elements.
(8) Parameterization: A measure of the use of a minimum of unnamed constants.
(9) Style: The appropriateness and use of standard conventions to aid in visual presentation of structures (e.g., numbering scheme, indentation of structures, blank lines between procedures and function definitions, and other factors which affect the readability of the software.
(10) Documentation: Availability, completeness, and correctness.
(11) Complexity: Degree to which module flow can be traced (typically measured using a McCabe's value).

S5.5.1 Proposal Evaluation Supportability Criteria

NOTE: Evaluation criteria will cover all requirements within the request for proposal (RFP), including computer resources development and management activities and the offeror's software management plans contained in the SDP and other applicable documents. The key to achieving supportability is by defining contractual processes, performance requirements, and metrics to which the contractor will commit and adhere during software development. These are important evaluation factors and must be included in the RFP.

- Availability of software, documentation, and rights necessary to meet life cycle needs.
- The compatibility of the proposed design with the support concept in the CRLCMP.
- For systems where software changes will be frequent and are critical to overall mission capability, quantitative criteria should be established to ensure the design is modifiable and proposed support resources and methods are adequate. The offeror should describe how to identify and reestablish a previous software configuration.
- When processing of sensitive or classified information is involved, ensure computer security is an evaluation criterion.
- Correctness and reliability (or their supporting criteria of traceability, completeness, error tolerance, accuracy, and simplicity) should be measured, over the entire life cycle, on every program.

S5.5.2 Other Evaluation Criteria

Other evaluation criteria should include:

- Throughput and memory capability of the proposed computer;
- Future vendor support for commercially supplied items such as tape drives, disk drives, and controllers;
- Computer resources interfaces to the rest of the system architecture and human operators;
- Adequacy of the operating system or software executive;
- Availability, currency, and usage of software development plans;
- Organic supportability of computer hardware and software;
- Offeror's software development plan and software development standards and procedures;
- Offeror's software development capability and capacity.
- Defect-removal efficiency (e.g., rate of 95 % or higher is acceptable).

CAUTION: defect-removal efficiency can be manipulated. Changing the definition of a defect from a development defect to a production defect can affect the metric.

S5.5.3 Source Selection Evaluation Considerations

MIL-HDBK-347 is geared to DoDD 5000 series documents and should be used in conjunction with top level software support guidance provided government-directed documents, which you should follow throughout the period of the contract to ensure a supportable and supported system is fielded. Not only should the attached factors be included in the Sections L and M of the RFP and the offeror's proposal, but they should also be incorporated as requirements in the Statement of Work. Much of the issues addressed in the following can be addressed in the offeror's SDP, therefore the Instruction for Proposal Preparation should require the offeror to submit a draft SDP for the system being acquired.

- CMU/SEI-94-TR-06 (This assessment is normally conducted by the offeror with SEI assistance before source selection occurs.) The Instructions for Proposal Preparation may inform the offerors that only those proposals from offerors who have received a Level 3 rating or higher will be evaluated.)
- AFMC Pamphlet 63-103 (Use of the tool requires a competent government team and a significant amount of time to complete, but it provides the program office an estimate of the level of risk that can be expected in the software development process using each offeror.)

NOTE: A SQM proficiency audit or SEI audit may also provide the desired assessment results.

Use of these tools will produce ratings in the following areas which should be reported to the source selection authority: program management, planning and execution, configuration management, quality assurance, quality measurement, training, process focus, and overall.

S5.5.4 DoD-STD-1467(AR), *Software Support Environment,* Considerations

Contractually specifying the exact SEE prior to initial development is not a good approach to acquisition. Instead, the offeror should specify the top-level requirements for the SEE, and the detailed implementation of this environment should be allowed to evolve.

- The offeror should identify any proposed software or documentation with limited or restricted rights. The offeror should identify any licensing agreements that apply to the software engineering environment or software test environment to be delivered to the Government. The offeror should describe how in-house personnel or a third party contractor can accomplish software support within constraints imposed by the rights and/or licensing agreements.

NOTE: Although not realistic to obtain in an offeror's response to a proposal, it is also desirable to know what data rights restrictions/licensing arrangements apply to the SEE/STE to be used by the offeror (or proposed subcontractors) and how those restrictions/arrangements will apply to the SEE/STE delivered to the Government.

- If the government's SOW requires the offeror to use the government's designated life cycle software support environment (LCSSE), have the offeror describe how the resources of the government's LCSSE will be used.
- The offeror should respond to the government's desire that the offeror's environment, identified in the proposal, shall be used in the subsequent contract performance, and that the offeror agrees to notify the Government of any changes required in the environment, with rationale given for the changes throughout the period of the contract.
- Have the offeror describe how all delivered products, both mission and support (e.g., products used in the factory development environment), will be integrated in and perform with the government designated LCSSE.
- Have the offeror identify the proposed sources for all software to be delivered in the SEE/STE.
- The offeror shall describe how the designated LCSSE might be used by the Government or the government's designee to evaluate, generate, install, integrate, test, modify, and operate the formally delivered software.

S5.5.5 AFOTEC Pamphlet 99-102, Volume 3, *Software Maintainability Evaluation Guide,* Considerations

- Have the offeror document the approach to ensure supportable software (i.e., the software has supportability characteristics), using the definitions in paragraph 4 of the basic document.
- Have the offeror provide samples of software documentation from a program of similar scope and effort and use evaluation procedures of this pamphlet to assess its supportability. Ask the offeror to identify any changes made since that software was produced which might be relevant to the current effort.

CAUTION: When this approach is used, you run the risk of not treating all offerors equality since some may not have a viable documentation base to be evaluated. Also, the evaluation of the documentation may not include the Government influence in the decisions and direction that led to the particular software and software documentation that was produced in the previous effort.

S5.5.6 Additional Considerations

- Have the offeror describe how the software development effort and costs will be visible to assure the effort is on track. The offeror's description of development/cost tracking should include a detailed explanation on the use of software work packages. The offeror should also be required to describe how the quality of software will be measured and maintained during the development process and over the life cycle.
- Have the offeror describe ways they will keep COTS items in their latest configuration and upward compatible to changes after delivery of the system without affecting system performance. The offeror should also address contingency plans for support of COTS products in the event the COTS vendor drops support or goes out of business. Also have the offeror describe how licenses and titles for COTS items will be transferred to the LCSSE.

NOTE: Decisions to upgrade configurations must be a joint decision between the Government and contractor, with the Government having ultimate control.

- Have the offeror describe how software functionality will be allocated, traced, and its quality measured between mission and system software computer software configuration items (CSCIs). Recognizing that specific design details are not set in concrete at the time of source selection, this information should describe what intrinsic system services (i.e., services contained in system software) will be needed and what the planned COTS utilization will provide these systems. By extension, the offeror can describe in general terms what non-system service functions (i.e., functions contained in mission software) will also be needed.
- Have the offeror describe procedures for performing independent verification and validation (IV&V) (if required in the contract) and ensuring the IV&V agent access to software and associated documentation. Also, have the offeror describe how duplication of effort between the IV&V agent and SQM agent will be avoided.
- Have the offeror describe how they will minimize/eliminate use of different type/manufacturers for processors used in the system, unless those different types would make use of existing workstations/ server resources at government operational and support locations.

NOTE: Limiting processor types is a specification issue, not a source selection issue. Source selection is not used to impose requirements.

- Have the offeror describe how compliance will be achieved with the requirement to deliver or provide government access to all software documentation (deliverable and non-deliverable) on electronic media or in digital format (i.e., paperless, computer-aided acquisition logistics systems (CALS) compliant). Include in "*Software Documentation*" software quality measurement data, including raw data, score sheets, tiered scores, problem trouble reports, and corrective actions.

NOTE: The Government should specify what form of electronic media is acceptable and have the offeror describe how compliance with that form will be accomplished.

- Have the offeror describe how software will be loaded into storage media. The offeror should document how and where software will be uploaded into the equipment (e.g., at what maintenance level (on/off-equipment), and with or without requiring removal of processors from the equipment) and describe the memory technology proposed (e.g., programmable read only memory (PROM), ultraviolet PROM (UVPROM), electrically erasable PROM (EEPROM)). Additionally, the offeror should provide the rationale behind the decisions made to determine the support concepts/maintenance levels.

NOTE: It is inappropriate to require the offeror to identify the memory technology planned for use.

- Have the offeror describe how specific critical design requirements (e.g., spare memory, timing, standardization of processors within system, etc.) will be met.

S5.5.7 Software Language Considerations

Have the offeror describe how they plan to comply with DoD 5000.2-R and DoDD 3405.1 software development language requirements. If they can not comply with these software development language requirements, the offeror must provide a rationale based on *life cycle* (and not just developmental) cost evaluation.

S5.5.8 AFSSI 5100, The Air Force Computer Security (COMPUSEC) Program, Considerations

- **Trusted Computing Base (TCB)**. Have the offeror describe how they will address each of the evaluation criteria set forth in DoD-STD 5200.28, *DoD Trusted Computing System Evaluation Criteria*, for the appropriate trusted computing base, depending on the sensitivity of the data and the clearances of the users.
- **Risk Management**. Have the offeror describe how they will address risk management requirements, including risk analysis, security test and evaluation, and certification for facilities, software development center processors, and embedded software used or developed under the contract. Systems must be accredited by the Defense Audit Agency before they are placed in use.

Have the offeror describe how all the automated computer security provisions (identification and authentication, audit trails, and file protection and control) will be met.

S5.5.9 MIL-STD-498 (or Industry Equivalent) Documentation Requirements Considerations

The offeror should address adequate completion of the appropriate documents listed (by DID title) in paragraph 6.2 of MIL-STD-498 (or industry equivalent), and describe how documentation adequacy will be evaluated.

S5.5.10 AFOTEC Pamphlet 99-102, Volume 5, *Software Support Resources Evaluation,* Considerations

- The offeror should describe their approach to addressing the software support environment (i.e., software support concept
- Software support resources should address the required personnel, support systems, and facilities required for supporting software during its life cycle.

S5.5.11 Other Supportability Source Selection Considerations

- Require that the offeror states conditions for software licensing. Specifically addressing the ability of the Government to process under a single site license with the right to copy for large quantity systems (e.g., Z-248 personal computers).
- All commercial-off-the-shelf software obtained for general purpose information systems processing equipment is required to be approved through the computer systems requirement board (CSRB) for management information systems.
- The offeror should provide data for applicable software, indicating any software attained under public domain libraries.
- The offeror should describe how contractor proprietary rights to proposed software will be minimized. While it may be difficult to control rights to commercial off the shelf or third part software, in-house developed software should be the property of the Government and be delivered as part of the life cycle software support environment.
- The offeror should describe the approach for transitioning the software process, products, and documentation to the supporting activity.
- The offeror should describe the approach for preparation, including training, of software support activity personnel for accomplishing the software support mission.
- The offeror should program PDSS personnel, facility, and equipment costs up front and include these in calculating system life cycle costs. Facility costs should include location, site preparation, construction, and installation.
- The offeror should make recommendations as to the optimum support concept (contractor only, Government only, or contractor/ Government mix) for each proposed computer software configuration item, and justify the recommendation based on operational requirements and life cycle costs.
- Have the offeror describe how the system/software engineering environment will meet all trusted database and multilevel security requirements.
- Have the offeror describe the level and sources of training (skills) required for support of each of the delivered software products.
- Have the offeror describe how the software will be implemented without serious impact to the operating system (if applicable).
- Have the offeror describe how the developed software will fulfill requirements and yet meet RFP interface requirements.
- Have the offeror describe how impacts of the newly developed software on other operating systems will be assessed.
- Have the offeror describe the strategy for reuse of existing and newly developed software.
- Have the offeror describe plans for software disaster storage and recovery.

S6.0 Tab 6: Lessons-Learned in the GSA Trailboss Course

George Coulbourn
Boeing
Barry Ingram
EDS

S6.1 Background

In 1988, the GSA introduced a two week course known as Trail Boss. The program was designed to train Government personnel in the conduct of information technology (IT) acquisitions. The original goal was to train approximately 300 people. After 10 classes this goal was met; however the demand continued, and in 1995, with the advent of class number 20, over 600 people have received the training. Beginning with class number 3, industry participation was requested through the auspices of ITAA, the Information Technology Association of America. The authors were privileged to conduct most of the industry sessions during this period. Many other members of ITAA participated in various presentations over this time period. The materials prepared for that purpose have evolved and have been used in numerous presentations by members of ITAA and by others. Most presentations have occurred in an interactive setting, such that problems and impediments could be openly discussed and analyzed. As a result of these experiences, Government and industry have learned many lessons. The purpose of this paper is to capture the more pertinent lessons in hopes that by their promulgation, both Government and Industry will benefit.

S6.2 Industry Trail Boss Presentation Approach

Fundamentally, the ITAA presentation is about mutual understanding and human and corporate behavior. We discuss processes and problems from the perspective of an independent systems integrator. We present a detailed description of how we make our procurement bid decisions. We describe the acquisition process from our perspective and discuss our objectives and concerns at each stage. In addition, the perspective of the subcontractor is presented and actual cases are discussed to provide real examples. We encourage and usually obtain interaction with the audience. Depending upon the audience and the time available, various topical issues may be addressed. Finally, lessons-learned by us are presented in the form of recommendations for consideration by acquisition teams, and the input of the participants is taken by the industry presenters for a better understanding of the Government's issues.

S6.2.1 Lessons-Learned

Following is a list of lessons-learned by the authors. They are a result of more than one hundred presentations, made together and separately, on the subject of IT acquisitions over the past five years. An appropriate disclaimer regarding the selection criteria, completeness, and presentation is hereby made: these lessons are presented in a format that could be presented as recommendations to a Government team delegated the responsibility of acquiring a large IT system. Some observations are controversial. All pass the authors' tests of being legal, achievable (albeit difficult) and mutually beneficial to Government and industry. No attempt is made to prioritize them.

S6.2.1.1 Obtain Top Management Support Before Proceeding

IT system acquisitions are difficult endeavors, at best, and impossible at worst. The process must conform to a host of laws, regulations, and policies that govern procurements in general, and then conform as well to another set devised exclusively for IT procurements. Occasionally, pressures from within the agency, from others in Government, or from industry can present obstacles that the acquisition team cannot overcome. Experience has shown that on almost all large acquisitions, there are times when success requires a tough decision by a senior executive.

The larger the procurement, the more players involved, and the longer the duration, the greater the potential for problems requiring executive action. When programs within agencies compete for funding and other resources, or when challenges to scope or other requirements arise, executive involvement is sometimes necessary. Furthermore, in a large, complex procurement, there simply are times when the authority to direct, countermand, or waive certain actions is essential to success.

Executive support should be obtained upfront. Obtaining executive participation and "*ownership*" should be an integral part of the acquisition strategy. The acquisition team should find an executive sponsor (or sponsors) and periodically review their acquisition strategy, milestones and risks. Care should be taken to highlight the major threats to success. The range of responses that might be required should be discussed to ensure that executive support accepts the exposure. Properly done, senior management is informed and ready to act when required. Finally, the Trail Boss program can help. Obtaining a "*Trail Boss*" designation from GSA requires a higher level of agency executive involvement than might otherwise be customary.

S6.2.1.2 Consider a Congressional Support Strategy

Good programs can die without Congressional support. The authors have seen this occur many times in the past with reduced Federal budgets and close scrutiny of all programs. It is important to maintain high visibility of programs to ensure continued life. This support must be consistent and must last throughout the acquisition and program phases. It is important, therefore, that a good Congressional support strategy be developed and maintained. This may take the form of frequent briefings of schedule, funding issues, program threats, technology requirements, and mission objectives to Congressional staff. Reviews of the potential savings and advantages of the program can be given to highlight the program's importance. Executive level support from Agency management is vital in sustaining Congressional support. In fact, it's their job.

S6.2.1.3 Involve Your End Users Meaningfully and Continually

The need to ensure end user involvement is so obvious that it might not warrant discussion except that, obvious or not, some acquisition teams fail to obtain it. Program success demands that the system be accepted by end-users and that, by their use, the system performance objectives are achieved.

It is not particularly difficult, in principal, to obtain end-user involvement. The most difficult steps are the first ones: identification of a representative sample of the end-user community and obtaining their commitment to support the acquisition. If these two steps are done properly, the probability of success is enhanced considerably. If not, the risk that the system may not perform as expected, or not be accepted by the end-users may be high. Roles for the end-user representatives include the following:

- Help define the system requirements,
- Assist in prioritizing requirements,
- Assist in defining "*mandatory*" and "*desired*" features,
- Ensure that the requirements are captured in the text of the RFP,
- Help mediate conflicting requirements within the user community,
- Continually validate their decisions within the user community,
- Help determine whether to incorporate changes in mission, policy or technology into the process,
- Participate in risk assessment and mitigation decisions,
- Concur with any changes made either to requirements or policy during the process,
- Provide the end-user perspective during interfaces with the bidders, especially during any demonstrations, and
- Help prepare the user community for the changes that the system will bring.

To perform these functions, end-user representatives should serve on the acquisition team and play a meaningful role in the evaluation and selection process. The challenge for the acquisition team is to ensure that the end-user representatives remain a representative sample of the end-user community throughout the process.

S6.2.1.4 Market to Your Vendors Pre-RFP

Vendors need to be brought into the acquisition process as soon as a need is established and while the requirements are being developed. By getting industry involved prior to issuance of the RFP, they can offer technological and business advice without jeopardizing the procurement, since this is prior to any formal documents being formulated and communications being restricted. In addition, new technologies and capabilities not previously known or understood can be considered as possible alternatives.

While the contracting community is competing against each other for your business, Government, in turn, is "*competing*" for the attention of qualified bidders. Since contractors' resources and bid and proposal funds are limited, enticing qualified bidders to consider the program is critical to the successful accomplishment of the acquisition. This time period provides a unique opportunity for both industry and Government to look at possible alternatives and solutions in an open, noncontentious environment.

S6.2.1.5 Develop a Plan to Use the RFC or DRFP Effectively

The objective of the Request for Comments (RFC) or Draft RFP (DRFP) process is to gather information to prepare an RFP which best reflects the real requirements and fulfills the needs of the end-user, and to prepare the vendor community for the coming competition. Therefore, development of a plan to utilize these vehicles most effectively is essential. Some of the specific goals of the plan should be the following:

- Improve the overall requirements definition,
- Include all anticipated sections of the RFP for a more complete review,
- Minimize questions and surprises after the RFP is issued,
- Minimize ambiguities in the RFP,
- Minimize delays and changes,
- Get recommendations on improving the RFP, and
- Attract qualified bidders.

Changes and improvements in the solicitation made at this early stage of the procurement process contribute to a much smoother process later on. Just as in a software development program, time spent on the front end of the effort to completely define and document the requirements and scope of the program results in lower overall costs and time expenditures. Conversely, the cost and time required to revise the designs and requirements after the RFP release are very high, both to Government and Industry. Changes later in the program may require bidders to adjust teaming arrangements, re-engineer solution designs, and even reverse previously positive bid decisions.

S6.2.1.6 Use Experienced Qualifiers

It is to the Government's advantage to get only qualified bidders. This is especially important on large contracts with high mission risk. Therefore, developing and requiring certain levels of experience or proven capabilities is a valid means for qualifying prospective contractors. This may take several forms:

- Past team experience on contracts or programs of similar scope and magnitude,
- Proven team capability in a particular technology,
- Proven software development capability,
- Documented software engineering maturity,
- Corporate size to absorb and compensate for risk inherent in the program,
- Adequate numbers of qualified staff with pertinent experience on the team,
- Proven program management experience,
- Capability to provide global support, and
- Capability to run a Live Test and Demonstration.

While these may be seen by some as limiting to competition, they are important criteria in the selection of any qualified team to ensure success. They do not prohibit smaller contractors from bidding, since they have the opportunity to become players on larger teams. In fact, this teaming may provide them access into some new areas. Furthermore, by clearly stating the qualifications expected of bidders, vendors can better gauge the appropriateness of preparing a bid and subsequent protests may be avoided. However, the Government should also be sure that any "*qualifying*" requirements are actually needed and provide a real advantage to the program. We have found that non-value-added requirements may eventually get removed from the final list of mandatory requirements and that they may have only added cost without benefit.

S6.2.1.7 Use Functional (Performance) Specifications

Historically, many sets of procurement specifications have been "*prescriptive*," meaning that RFPs ask for specific products or products with specific hardware capabilities: typically a commodity type of product, for example a video display device with a 1024 X 768 pixel resolution. Another form of the prescriptive requirement is to prescribe exactly how you want a service performed, rather than the end result of the service. While this may be the preferred method to acquire commodity products, it is very constraining when the procurement is for large or complex systems. Also, when the procurement duration is lengthy, prescribed products may become outmoded or may be overtaken technically by superior offerings. In these cases, vendors may be unnecessarily limited in selecting products or attributes for their solutions.

The Government is usually best served by providing "*functional*" specifications which describe the requirement or need, and ask for a solution from industry, without specifying the actual methods, products, or technologies to be utilized. With this freedom, industry has the opportunity to offer new ideas and is not

constrained by a particular technology. We realize that asking for functional requirements may make the evaluation more difficult, but the reward of a better, more current, or more effective solution may justify the increased effort.

S6.2.1.8 Challenge Complexity and Non-Value-Added Reviews and Approvals

Most enterprises knowingly and unknowingly support non-valued-added activities. In industry today, considerable effort is being expended in eliminating work that does not add value as "*perceived by the customer*." Internal administrative work is important when it enhances the quality of the product. It keeps decision makers and supporters informed, but most such work does not meet the value-added test. Worse, it adds expense and diverts talent from other tasks, and worst of all, it increases the duration of the process.

Reducing non-value-added work requires that the acquisition team challenge the culture of their organization. The challenges are to written and unwritten policy. Experience has shown that the challenges can usually best be made at the start of the acquisition process, when the team is laying out its schedule, milestones, and developing its rules.

An approach that has met with some success is to define the acquisition approach such that reviews and approvals are combined, conducted in process, or eliminated altogether. The ability to do this is considerably enhanced if senior executive support has been obtained, and is facilitated if the executive has '*bough- in*" to the concept.

S6.2.1.9 Distribute Risk Equitably

Contractors devote substantial time to risk analysis regardless of whether the RFP explicitly calls for such analyses. The reason is that all risk that must be borne by the contractor must be identified and planned for to ensure that an realistic business case is developed. Contractors must either price or mitigate all elements of risk. The RFP is the basis for the risk analysis since it furnishes the bidder with the Government's apportionment of risk. During the acquisition phase, risk analysis and pricing by bidders is a business decision to which the Government is a party.

Every risk that the Government lays off on the contractor (that the contractor accepts) will increase the price from a responsible bidder. Hence, the Government itself should analyze each element of risk to determine, first, whether it can be managed by the Government; and then, whether it is more cost effective for the Government to assume the risk or pay for the contractor to do so. Even in instances where the Government cannot manage an element of risk, it may be less costly for the Government to assume it anyway if uncertainty will cause the bidder to assign a high cost to the risk element. Indeed, in the extreme case where neither the Government nor the contractor can manage an element of risk, the Government should always assume it; otherwise, the Government pays a premium for a service that cannot be performed and risks contract disputes. A lesson learned and relearned is that attempts by buyers in all areas of society to lay off unreasonable risk on their contractors backfire. Invariably, when difficulties occur, cooperation erodes, the contractor looks for a means of escape, the mission suffers and both sides lose.

Time spent by the Government, especially during the draft RFP stage, in understanding the potential bidders' perspectives concerning risk may be the most useful effort of all. Only by talking with potential bidders can the Government expect to understand how the bidders assess the risk. And, it is the bidder's assessment that matters at this point.

S6.2.1.10 Really Work Hard on Internal and External RFP Integration

Internal integration means assembling an RFP in which the Sections (especially Sections C, H, L, M and the Technical Specifications) are consistent with each other. This is difficult to do because of policies, and sometimes law and regulation, that mandate inclusion of contract clauses unrelated to user and system performance requirements. It is also difficult when agencies prepare prescriptive specifications (i.e. specifications that prescribe elements or products, rather than performance requirements).

It is difficult to overemphasize the importance of this point. Ambiguity is the acquisition team's enemy. Ambiguity allows different interpretations by bidders, and bidders are generally entitled to the minimum interpretation. Ambiguity can make performance evaluations and tests difficult or impossible. It can precipitate delay when amendments are required for correction.

Because the various RFP sections are typically assembled by different teams, integration among the sections is a separate activity that must be planned *before* the sections are prepared, and completed *afterwards*. The planning starts with the recognition of the need and the complexity of the task. Care must be given to the preparation of guidelines and standards for RFP section writers. Interim reviews are important. Independent reviews can also help. Electronic tools for RFP decomposition or "*shred*" (employed routinely) by bidders can assist in expediting the task of comparing families of requirements. All of the above takes time and time should be allowed in the schedule.

External integration means assembling an RFP that is consistent with the overall acquisition strategy, the mission of the system, the mission of the agency and realities in the agency's internal and external environments. It is also concerned with establishing the kind of relationship desired between the agency and their contractor. The apportionment of risk between agency and contractor is critical to the contractor's behavior. Hence the contract type, evaluation criteria and methodology, mechanisms for change and technology refreshment, performance measures, and penalties, if any, must be dealt with consistently.

S6.2.1.11 Publish Detailed Evaluation Criteria and Methodology

There are two powerful reasons for publishing full and complete evaluation criteria and for doing so as early as possible (draft RFP stage): first, to enable bidders to know what is really important to the buyer; and second, to enable the agency to avoid protests.

Why would an agency not want its bidders to know how it will decide from among its offers? Detailed evaluation criteria allow bidders to design their solutions in accordance with what really matters. Denying bidders the detailed information forces them to guess (*and they will guess!*). Accordingly, the process may favor bidders who guess more accurately, although they may have no more real knowledge of the importance of the criteria. Likewise, clearly stating the relative importance of detailed evaluation criteria will also ensure that the most important requirements are adequately addressed in the solutions offered.

Perhaps, the most effective approach to avoiding protests is to furnish detailed evaluation criteria, follow the criteria meticulously, and debrief bidders in detail in accordance with the published and practiced criteria. When bidders understand why they lost (and hence, why the successful bidder won), and believe that the decision was made fairly and in accordance with the evaluation criteria, the principal reasons for protest are neutralized. Published, detailed evaluation criteria allow the agency to accomplish this. However, it is necessary that the agency plan for this process from the start. In other words, the agency must recognize that the evaluation process has two products: the selection and the debriefings of the unsuccessful bidders.

S6.2.1.12 Ensure that the Evaluation Methodology and Criteria both ALLOW and REQUIRE You to Select the Best Vendor

Obviously, the evaluation criteria, along with the technical specifications in the RFP, drive the solutions proposed. Therefore, it is incumbent upon the Government to evaluate the proposal in accordance with the methodologies and criteria stated in the RFP. Any deviation from these procedures, without good explanation and reason, will invite questions and invariably, protests. This may suggest that adequate thought was not put into the evaluation process, or worse yet, favoritism is being shown to another bidder.

It is just as important that the evaluation methodologies and criteria enable the selection of the proposal and bidder that best meets the Government's requirements for the benefit of the program. Any criteria that force a selection other than this need to be removed. Along this line, it is important that only the necessary requirements are specifically stated in the RFP. Extraneous requirements or standards that provide no real added-value to the procurement or add unnecessary complexity should be avoided.

Detailed evaluation criteria are also a valuable defensive tool for the acquisition team in maintaining stability in the event circumstances change. It is not unusual in a lengthy acquisition, for missions, technologies, or people to change such that pressure mounts to change the acquisition strategy without changing the RFP. In such instances, the evaluation criteria can become a welcome constraint for the evaluation team, requiring them to stay the course.

S6.2.1.13 Tell your Bidders Everything

It is axiomatic that the more that bidders know about a customer's requirements, selection criteria and the environment in which the system will operate, the more closely they can design a system to met those requirements. Agencies penalize themselves when they withhold information that might materially affect design decisions made by a bidder. Bidders need to know not only the agency's best estimates of performance requirements, but also how those requirements relate to each order to be able to conduct meaningful trade studies during the design process. Trade studies involve not only technical designs, but management system operations and cost as well.

If information is not available, bidders will develop their own estimates. This can significantly increase the risk of wasted effort on the part of both bidders and the agency and increase the probability for protest. Most information that is denied bidders results from internal agency policy rather than law or regulation. For example, there is legal prohibition against providing bidders with cost data so that they can "*design-to-cost*," a common commercial practice. When this is done, most bidders will try to maximize their offering within the anticipated available funding, or bid somewhat less than the available funds to provide an attractive price. The advantages of "*designing-to-cost*" are many; solutions that meet yearly budget allowances with implementation plans in step with Government needs; more bidders within the "*competitive range*" from a cost standpoint; and, in many instances, innovative proposals offered which meet the requirements at substantially reduced costs.

Many mechanisms are available for information exchange, even after RFP release. Agencies need only to be careful that no bidder receives information not made available to all. In fact, the more information that is released to the entire bidding community, the better the quality and quantity of solutions that can be offered and the closer they should respond to the real needs. In addition, as more information is made available, there is less opportunity for unscrupulous individuals to attempt to provide or gain undeserved advantage. Prior to RFP release, and especially in the early stages of planning for an acquisition, agency and prospective bidders benefit from open exchange of information.

S6.2.1.14 Don't Drop the "Curtain" Until RFP Release

IT system acquisition is a lengthy and complicated process; in many instances, unnecessarily so. However, until fundamental changes are made, bidders and agencies will have to live with it. The key is not to make the system any more complicated or difficult than it already is.

There may or may not be a single root cause for the difficulty and complexity of IT system acquisitions, but one thing is clear: the very long duration of the process exacerbates all potential problems. It is the long duration that allows product cycles to render agency requirements definition and bidder solutions obsolete. The long duration provides time for agency needs, missions, and environments to change, thereby invalidating requirements. Over time, people and policy change. Yet communication is cut off, sometimes for more than a year before the agency selection process is completed.

Agencies that make their people available for information exchange with prospective bidders as long as legally allowed will benefit most. Unfortunately, some agencies cut off communication well in advance of RFP release; some before release of their draft RFP. This penalizes both bidders and the agency. And, it is unnecessary.

S6.2.1.15 Communicate with Your Vendors Frequently After the "Curtain" Drops

After the RFP is released, and sometimes even after the RFC is issued, the Government severely restricts communications with the vendors, and then usually only in written form and through the contracting officer. This is understood, although not appreciated, by the bidders as a way to prevent inappropriate discussions and prevent advantages to some bidders. This pattern of communication is reflected in Figure S-1 and compared with the communications in the commercial environment.

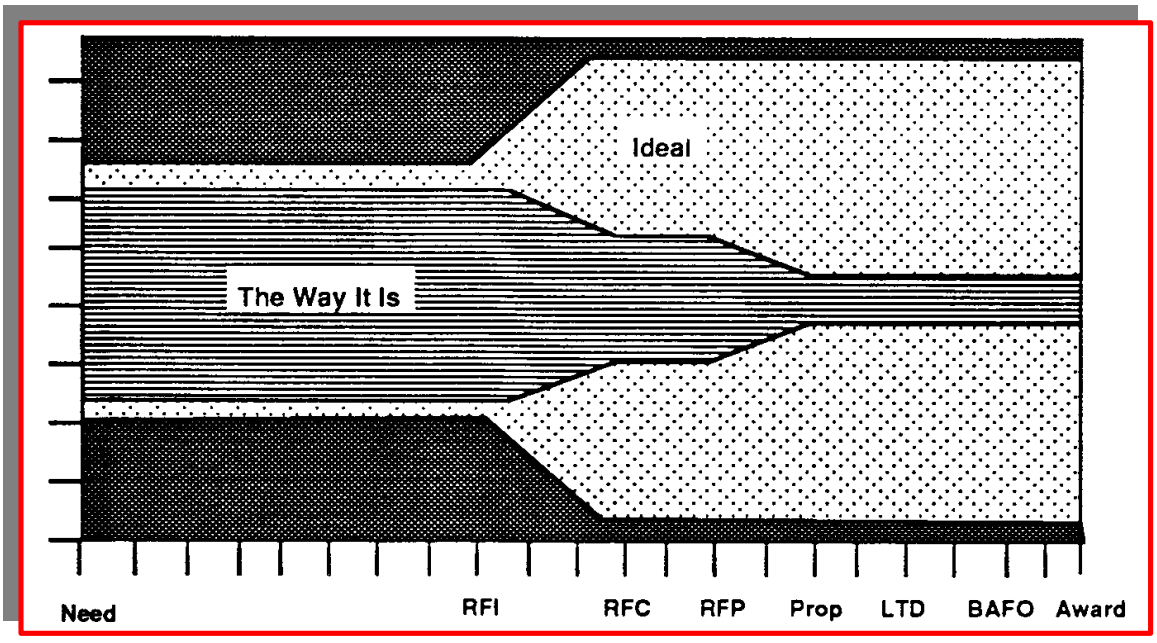

Figure S-1. Level of Communications — Federal versus Commercial

Notice how, in the commercial arena, communication actually increases as the program progresses. However, it is also important to keep communications open in the Federal procurement process with the entire bidding community after the "*curtain*" drops. It is essential that bidders feel that the program is moving, is under control, and is being pursued with as much enthusiasm as possible by both the program shop and the contracting office. One of the best indicators of progress is frequent communications about the status of the program, questions and answers, and accurate milestones and event dates.

As the bidding community reviews the programs they are bidding, those with poor communications may be dropped, simply because the contractors feel they are not as critical to the Government as they are to the bidders. Internal struggles within bidders' organizations for continuing funding to bid programs are common. As programs drag on for months and even years, good communications of accurate information about status is critical to preserving both the prime and subcontractor teams.

Finding ways of maintaining communications serves the Government in other ways. For example, in any given acquisition there may be bidders who are incumbents or are otherwise serving the customer through other contract vehicles. If other bidders do not have a presence in the customer's business, they may be (and certainly believe themselves to be) at a continuing, sometimes growing, disadvantage regarding access to changes in the customer's missions, problems, and preferences.

S6.2.1.16 Use LTDs Only to Reduce Unacceptable Risks and Uncertainty

Live Test Demonstrations (LTDs), or Operational Capability Demonstrations (OCDS) are expensive, time consuming, and difficult for both the Government and the bidders. They are also essential in some procurements. At times, they are necessary and at other times, they are superfluous. When large, complex systems integration or development programs are to be undertaken, it may be necessary to require an LTD. In these cases, it is important to be certain that the bidders' solutions, their methodologies, and their capabilities are closely examined in a real life scenario simulating the solution offered.

The capacity of a solution to perform at a given level may need to be tested, especially if the solution is offering new state-of-the-art products or are proposing products to be used for the first time. It may also be necessary to have the bidder demonstrate the corporate resources available to respond to unexpected events and schedule and quantity changes. Therefore, an LTD is a way of legitimately evaluating and screening bidders' capabilities to perform.

Some LTD's may not be necessary or offer increased evaluation knowledge. This scenario is most often the case when a procurement asks for an LTD for simple commercial-off-the-shelf (COTS) products. In this case, the technical specifications may be so well advertised and known to the IT industry for these products that the LTD delivers no added-value to the procurement or evaluation. It will, however, add substantial costs to both Government and industry. In other, simpler words, require an LTD only when one is necessary and can substantially reduce program risk.

S6.2.1.17 Be Certain that COTS and NDI Products Exist When You Call for Them

The requirement for COTS and NDI products is a very cost-effective movement within Federal Government procurements. Sometimes, however, the Government asks for products as COTS or NDI items that are not currently available to the general market in this manner. This may be because of misunderstanding on the part of the Government, overstatement on the part of vendors, the failure of vendors to offer these desired products by the time the RFP is issued, or simply because they are not available.

As vendors demonstrate products or discuss capabilities and features, COTS may be an assumed status. Many times these products are then specified in the RFP; and it may be impossible for the bidders to provide these as COTS or NDI, although they are in the development labs or offered to specific markets. In these cases, the bidders spend a considerable amount of time trying to get the products offered or responding to the Government that the products can not be supplied. Likewise, the time and expense to the Government in addressing or defending their understanding of the availability of these COTS or NDI products to the bidders is considerable.

S6.2.1.18 Integrate the Debriefing Requirements Early into the Acquisition Process

The acquisition process should produce two products: a selection and debriefing for the unsuccessful bidders. The debriefing activity should have two objectives: to tell the unsuccessful bidders why they lost, and to avoid a protest. To maximize the effectiveness of the debriefing activity, the acquisition team should plan for it as an integral part of the acquisition process, beginning early.

Too often, an acquisition team prepares for the debriefing late in the evaluation process, when it is completed. Preparation for a debriefing should begin in parallel with preparation for an award; that is, both should be considered, and given equal weight, when the evaluation criteria are prepared for RFP Section M. Work by the evaluation team should be captured as the evaluation process proceeds, for both the selection and the debriefings.

The same level of diligence and the same considerations should go into selecting a bid as in debriefing one. The criteria apply equally, the evaluation methodology is conducted equally, and the results are weighed equally. The evaluation criteria are used both to make a selection and as the basis for the debriefing. (In fact, if criteria other than those in Section M are used for either, a protest is invited.) The first and most effective line of defense against a protest is for the agency to tell the unsuccessful bidders why they lost and why the successful bidder won, *in detail*, against the published evaluation criteria. That objective is facilitated by the approach outlined above.

S6.2.1.19 Cut to a Small Competitive Range When Possible

Preparing a large IT system proposal is expensive and requires often scarce resources. Most bidders begin the process believing that their solutions will be competitive and expecting to be selected. Disappointment at not being selected is natural. However, all bidders would prefer to be eliminated from the competition immediately following the determination by the evaluation team that their proposal will not be selected. Better to cut your losses and redirect your efforts than to waste another unit of a scarce resource.

The agency also benefits significantly from a cut in any procurement involving four or more bidders because they, too, can save scarce resources, concentrate on the most qualified bidders, and shorten the evaluation period. Unfortunately this process is not often chosen. Either the agency fears an immediate protest and attendant disruption, or it is not prepared to make and defend a cut.

The best approach is to plan from the start to be able to cut to a competitive range in the event two conditions are met: a sufficient number of bids are received, and rankings among the bids are sufficiently different. The acquisition strategy should incorporate the plan. Bidders should be informed of the intent. Evaluation criteria should be sufficiently detailed to support comprehensive debriefings. Senior management should be informed and ready to support.

S6.2.1.20 Have Oral Presentations and Discussions

There are two major benefits in holding oral presentations and discussions. These are:

- Potential to better understand the offered proposals, and
- Opportunity to meet face-to-face.

It is often difficult to understand the full capability or advantages of a proposal by only reading the submitted document. With a full oral briefing of the document and the solution, you have the opportunity to have more in-depth discussions of the details of a solution and raise questions about specific areas, without waiting for time-consuming written questions and answers. These presentations and discussions can be followed with written exchanges to document the sessions.

Orals and discussions give you the opportunity to see your potential providers face-to-face before you award a contract to them. You have the chance to see them in action, how well they interface with you, how they respond to issues, how they may address problems after award, and how comfortable you are going to feel with them. Granted, much of this is subjective, but since you will work with the winner for several years, the process will be of value. It will give you the opportunity to meet the key players and some of the executives responsible for the program from the bidders' corporations. Further, if you combine the presentations with site visits, you can view the corporate capabilities and the depth of personnel available to be called upon to assist in your mission.

Oral presentations give the bidders an occasion to discuss their understanding of the important issues and the mission of the program office. It may also be an opportunity for you to meet the entire bidding team, the prime, and the major subcontractors and teaming partners, to investigate the strength of the entire team, and to view their commitment to the program.

S6.2.1.21 Schedule Submission of the Cost Volume at Least 2 Weeks After the Other Volumes — More if No BAFO

Bidders prepare their proposal sections and volumes at different paces. Commonly, they conduct trade studies, make architecture decisions, and complete designs prior to proceeding with other steps. Sometimes multiple iterations among options are required. Cost estimating necessarily follows solution definition. Proposal teams lay out detailed schedules to manage the completion of design work, pricing and estimating, and incorporation of the work into text and graphics.

Necessarily, completion of the cost estimates, supporting rationale, and the incorporation of these data into the Cost Volume, must await the completion of all other portions of the work. Hence, bidders set interim (internal) completion dates for non-cost elements to allow sufficient time to complete the Cost Volume. Pressures to perfect the solution, sometimes incorporating late-arriving information (sometimes from the agency) makes it difficult for the bidder to hold to their dates for completion of the non-cost volumes sufficiently in advance of proposal due date to allow time for orderly assembly of the Cost Volume. Even though this may be "*the bidder's problem*," if the agency can significantly mitigate the problem (and it can), all parties benefit.

The agency that recognizes the reality of this problem serves itself by scheduling the submission of the Cost Volume at least two weeks *after* the other volumes. Invariably, the product will be significantly better. It

will contain fewer errors and be easier to review. It may represent a lower price if the additional time allows bidders more opportunity to negotiate prices with subcontractors based on the completed solution.

In the special case where an agency considers awarding on the basis of initial submittals, this approach is especially important. It may even be the key to making the no-BAFO strategy work. The reason is that to award without BAFO, the agency must forego the opportunity for discussions and revisions to proposals. Hence, the more complete and accurate the initial (only) proposal, the better for the agency. Allowing even greater time in this case warrants consideration. The best approach is to ask the bidders how much time they feel is required, during the DRFP period, decide, and stick with that decision.

S6.2.1.22 Find a Way to Waive Cost or Pricing Data Requirements

What are the real requirements for cost or pricing data? The Federal Acquisition Streamlining Act (FASA) amended the Truth in Negotiations Act (TRNA) to reduce the requirements for cost or pricing data. If a specific exemption applies, the contracting officer no longer has the discretion whether to require the data. The threshold for submission in civilian agencies has been raised to $500,000 and DoD's temporary threshold of $500,000 has been made permanent. Data requirements have also been relaxed under the rules for acquisitions of commercial items.

S6.2.1.23 Critically Examine the Need for 3rd Level and Lower Detail in the Cost Volume (the Lowest Value-Added Exercise of All)

There is typically more cost information required in a proposal submission than can be read or verified in any reasonable amount of time. Bidders are asked to supply this information in both paper and electronic form. Because of the detail desired, and the concern to know all the possible information about the costs, there are instances where following the requirements stated in the RFP would result in 30,000-plus pages of cost information. This volume of data could never be utilized. In addition to increasing the cost of the proposal efforts and the Government's cost to attempt to verify the data, the process adds measurably to the procurement duration. Since we are developing information that will not be utilized, we are paying for a non-value-added activity.

Finally, with greater detail, more errors are inevitable because of the increased complexity of cost element reconciliation and the "*time crunch*" that occurs as the Cost Volume is assembled during the final days of the process.

S6.2.1.24 Allow Enough Time for the Intense Activities

After "*Lack of Communications,*" the second most common complaint heard from bidders concerns time extensions to proposal due dates. Bidders, in general, prefer for the agency to allow enough time for proposal preparation, and then to stick to their schedule. The practice of allocating insufficient time for proposal preparation, then granting an extension is highly disruptive to responsible bidders. Too many acquisition teams employ a strategy of allowing insufficient time knowing that an extension will be "*necessary*" and intend to grant one or more. Of course, bidders bear a large share of the blame because it is frequently they who demand the extension. However, extensions punish the bidders who take the schedule seriously and reward those who do not, and it is the agency who grants or refuses the request. The preferred scenario would be to allow sufficient time for the preparation, and provide for no extensions, except when a major, program-threatening flaw is detected. Just say "*No!*"

The problem is that a complex proposal can require four or more months for assembly, and agencies have difficulties accepting such a lengthy period. The best approach is to ask bidders how much time they will require as part of the DRFP process and allow a little more. Adherence to the schedule also requires preparation and discipline on the part of the agency. The agency must avoid providing grounds for an extension by avoiding RFP amendments. Easily said, of course, but it is frequently possible to avoid material amendments if the DRFP process is conducted effectively. Effective DRFP processes can produce well integrated RFP's and no surprises for the bidder community.

Discussions with agency acquisition people consistently have shown a general lack of appreciation for the complexity and amount of time required for BAFO preparation. They frequently do not understand why four to six weeks or more are required. "*You knew that Call For BAFO was imminent; why weren't you prepared?*" Two conditions cause this situation. The first is that BAFO information is the most sensitive data that a bidder will ever collect. The second is that until Call For BAFO actually is given, the date may slide. Furthermore, even if the bidders are prepared, the process still requires considerable time.

The bidder's BAFO assembly process involves the incorporation of any changes required or allowed into the proposal and the repricing of the proposal. Repricing requires revised estimates by the prime bidder and solicitation of estimates and quotes from suppliers, frequently multi-tier. Estimates and quotes are revisited to negotiate the most effective distribution of risk and to obtain the most competitive overall life cycle price. Multitier corporate approvals may be required and several iterations may be necessary to obtain the final price.

BAFO prices represent the best proposal that the bidder can assemble. Because of the lengthy acquisition cycle, original proposal details can change materially. In a highly competitive environment, details of competitors' solutions cannot be protected for extended periods because of the mobility of suppliers and the number of different relationships formed and dissolved as bidders form other teams on other competitions, and people change assignments. Accordingly, bidders avoid collecting final pricing data prematurely.

S6.2.1.25 Never, Never, Never Slide the Schedule

When this statement is made in discussions with acquisition teams, the reaction is usually strong and emotional. However, when case studies of procurements (some very complex), that were completed on reasonable schedules without slides are discussed, certain patterns emerge. No one can guarantee a process for adhering to an IT acquisition schedule, however one deceptively simple observation can be made: "*The way to stay on schedule is to never let it slide*."

The recipe for success contains ingredients of varying difficulty to obtain. First, the schedule must be reasonable. The portions requiring bidder participation (especially proposal and BAFO preparation) should be established in consultation with prospective bidders. The DRFP period is optimal for establishing these dates. Analogously, time periods for internal acquisition team work elements must be negotiated carefully. The schedule should be published and emphasized frequently and emphatically. Top management support should be pre-arranged and should be prepared to overrule or waive challenges and back the acquisition team. Cutoff dates for internal and bidder interfaces should be set and enforced. Most importantly, no substantive amendments should be issued; they invite schedule slides, and for legitimate reasons.

How can an acquisition team conduct a procurement without issuing substantive amendments? By developing an RFP that is internally and externally well integrated and that, by effective use of the DRFP, contains no surprises for the bidder community. Hence, one that requires no substantive amendments. Finally, when an

acquisition team overcomes serious threats to its schedule a few times, everyone else (including the bidders) begins to take the schedule seriously. No one wants to be the cause of compromising it (and bidders can't afford to take a chance).

S6.2.1.26 Carefully Analyze, Map, and Monitor All of the Stakeholders

The Federal IT acquisition process contrasts sharply with commercial processes in several ways. The most dramatic is the lengthy duration. Next, is probably the large number of players on the Government side. For various reasons, some having to do with protecting the public trust, authority is dispersed and shared among a number of players. Identifying the distribution of authority and understanding people's objectives and needs is critical to success in any endeavor, and certainly so in an IT acquisition.

All organizations have both formal and informal structures. An IT acquisition has, at a minimum, its leader, technical support, contract management, the end-user community, and senior management. In addition, other agencies and elements of the Executive Branch, Congress, associations, and individuals may be directly or indirectly affected by the procurement.

Some of these people have formal roles and are easily identified, others are not. Various forces build and subside during the annual political cycles as budgets and missions are scrutinized. Temporary coalitions arise and fade. People, missions, and external pressures change during a lengthy acquisition. Any large successful endeavor requires strategies to deal with all of these factors. Strategies must be sufficiently robust and agile to recognize and address threats and react appropriately. Deliberate mapping and analysis of the entire community of stakeholders are essential to success.

S6.2.1.27 Challenge the Unproductive Things that Your Culture Requires

Most are NOT requirements of the FAR ("Most of the provisions of Federal Acquisition Streamlining Act (FASA) were not prohibited by the FAR")

Notwithstanding changes to laws and regulations, both Government and industry recognize that cultural changes must take place in parallel. Many people advocate that the Government lead this change. It must accompany the regulatory changes for procurement reform to succeed. OFPP issued new guidelines to encourage Government contracting officers to use personal choice, business judgment, and plain common sense. In one specific instance, OFPP reversed a long-standing belief in the acquisition community that anything not specifically addressed in the Federal Acquisition Regulation (FAR) is prohibited. The new focus is on what is in the best interests of the Government. Items to watch for (that may be considered nonproductive) include requiring standards that are not applicable, requiring documentation that will not be used or will be superfluous, and requiring too finely-detailed cost information.

S6.2.1.28 Objectively Evaluate and Share Acquisition and Program Successes

IT system acquisition is so complex that when the successful bidder is finally awarded the contract and given their notice to proceed, a great feeling of accomplishment pervades. Regrettably, a successful acquisition does not assure that a successful system will be fielded. Similarly, a troubled procurement does not doom the system being acquired to failure.

Curiously, in the Federal Government community, and in dramatic contrast to private industry, far more attention and scrutiny are typically given to the conduct of the acquisition than to the subsequent performance of the system itself. It is relatively rare for the successes, failures, and lessons-learned after award to be addressed, unless the system provider encounters major difficulties in fielding the system. One reason for this is that it may be several years after award before the results of new systems and processes can be evaluated. Another, of course, is that favorable news does not "*capture the headlines*." Indeed, several Government and Industry attempts to analyze whether acquisitions substantially met agency missions requirements in recent years have met with mixed success.

This culture is harmful to all. If it could be changed such that successful acquisitions and successful systems were dissected, analyzed, and reported, valuable information should result. For maximum effectiveness and credibility, such analyses probably must be conducted by Government personnel (as opposed to industry) who were not directly involved in the acquisition or system development processes. One approach might be to require agencies to prepare a formal report of the acquisition as a final step in the process and establish a "*clearing house*" or library of lessons-learned.

S6.3 Conclusion

As indicated earlier, the above "*lessons-learned*" were derived from the preparation and conduct of interactive presentations made by the authors and other to more than 1,000 government middle and upper managers during the past 5 years. Accordingly they are the "*intellectual property*" of us all. Many of these practices are now routinely followed by some agencies. Others are under consideration or under trial. Given the great diversity among agencies, there are some government managers who challenge whether some of these practices are allowable. However, we firmly believe that none of the above require changes to anything other than internal agency policy.

We also firmly believe that all of the above practices pass the test of being beneficial to both government and industry. Each item is worthy of consideration by itself, and taken as a whole they aggregate to a partial set of best practices given the current state of the art of IT acquisition management. As we continue to interact, there will be many more good ideas debated among us. The authors hereby solicit any comments in any form (criticism, corrections, additions and the like) from any interested person.

S7.0 Tab 7: Contracting for Success

Jerome S. Gabig, Jr.
Venable, Baetjer, Howard & Civiletti

S7.1 Abstract

Senator Cohen's scathing report, *"COMPUTER CHAOS: Billions Wasted Buying Federal Systems,"* alludes to "*inevitable problems with software development*" that cause cost overruns and schedule slippages. Cost overruns and schedule slippages need not be "*inevitable*." This presentation focuses on two critical success factors that enable the government to greatly increase the probability of a successful software development contract. The first critical success factor is equitably allocating the risks between the parties. The second critical success factor is structuring the evaluation criteria to maximize the probability of selecting the best qualified offeror.

S7.1.1 Overview

Where an agency must use a vendor to perform a software project, there are two critical success factors regarding the contracting process that greatly increase the probability of a successful software development effort. The first critical success factor is to structure the contract to allocate equitably the various risks between the parties based on which party is best able to manage the risk. The second critical success factor is to structure the evaluation criteria to maximize the probability of selecting the best qualified offeror.

S7.2 The Importance of *"The Written Word"*

A congenial relationship between the contractor and the government is almost indispensable to the successful completion of a software development effort. One might think that a congenial relationship would diminish the importance of the "*written word*." Instead, by minimizing the probability of misunderstandings, a well-written contract is a major contributor to a congenial relationship between the parties. As recognized by the Software Technology Support Center, "*once the initial contractor enthusiasm is over, the written word...has the most influence on contractor actions*." Experience has shown that when "*the written word*" is unambiguous as to the duties and responsibilities of the parties, the animosity that arises from quibbling over performance obligations usually can be avoided.

S7.3 Structuring The Contract to Best Allocate Risks

DoD has consistently recognized the need to structure contracts to allocate risks in an equitable and sensible manner:

> *"The contacting approach selected for each acquisition phase must permit an equitable and sensible allocation of risk between Government and industry."* [DoD Directive. 5000.1, Feb. 23, 1991, at C.3.]

Risks essentially fall within three categories: cost, schedule, and performance. Each of these three categories of risk deserves a separate discussion.

S7.3.1 Cost Risks

The foremost way of allocating cost risks is through the selection of the type of contract. For instance, FAR § 16.103(b) states that a firm-fixed price contract "*shall be used when the risk involved is minimal or can be predicted with an acceptable degree of certainty*." Although rarely followed by contracting officers, the FAR also admonishes that the contract type generally should be negotiated with the offerors:

> *"Selecting the contract type is generally a matter for negotiation and requires the exercise of sound judgment. * * * The objective is to negotiate a contract type and price (or estimated cost and fee) that will result in reasonable contractor risk."* [FAR §16.103(a)]

The reasonableness of the cost risk to the contractor is a factor of how accurately the contractor can estimate the cost to perform the work. The highly recognized work of Barry Boehm, as shown in Figure S-2, reveals the increasing degree of accuracy for estimating costs as a software development project proceeds through the phases of the waterfall model. Superimposed under the x-axis of Figure S-2 is the linear progression of contract types in the sequence in which they represent decreasing risk to the contractor. The superimposed x-axis should not categorically dictate the contract type for any particular phase of a software development project. Nevertheless, Figure S-2 correctly suggests that as the relative accuracy of the cost estimate increases, it is appropriate to select a contract type that correspondingly places increased cost risks on the contractor.

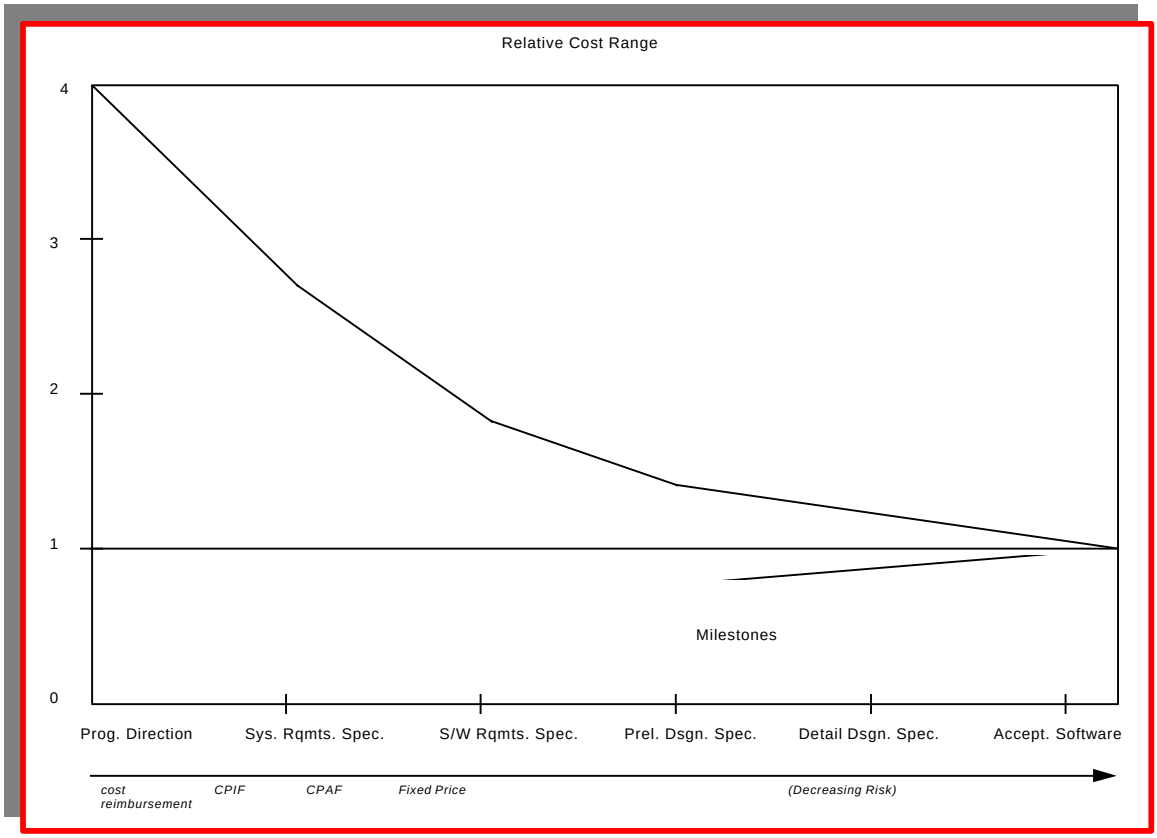

Figure S-2

S7.3.2 The Premature Use Of A Fixed Price Contract Invites Failure

Within the government, there has been a long-standing aversion to cost-reimbursement contracts because of a lingering suspicion that contractors are not motivated to work efficiency. Another suspicion has been that a contractor is less likely to assign its best software engineers to a cost-reimbursement contract. As a norm, fixed price contracts for complex software development contracts are not conducive to the iterative nature of the process. The following excerpt expresses a user's perception of the anticipated "*give and take*" necessary to refine the software requirements specification:

> *"Actually, the software specification review is an iterative process with the iterations consisting of the contractor submitting a draft of the spec, the technical monitor reviewing and recommending changes to the draft, the contractor making some changes and resubmitting a revised draft. The iterations continue until the program manager feels that the software requirements specification establishes the allocated baseline for its CSCI."*

Yet, on a fixed price contract, the contractor is apt to regard anything beyond the second iteration as unwelcomed meddling by the user's technical staff. Such behavior by the government's technical representatives may result in the contractor submitting claims. A fixed price contract that is inundated with valid claims typically does a poor job of shifting cost risks to the contractor.

S7.3.2.1 The Premature Use Of Fixed Price Contracts Favors Vendors With Immature Processes

A major difference between a vendor with mature processes and a vendor with immature processes is the consistent capability to accurately estimate the costs to perform a software development project. Figure S-3 graphically depicts the relative accuracy of a cost estimate by a SEI Level 1 vendor in contrast to a cost estimate by a SCE Level 3 vendor.

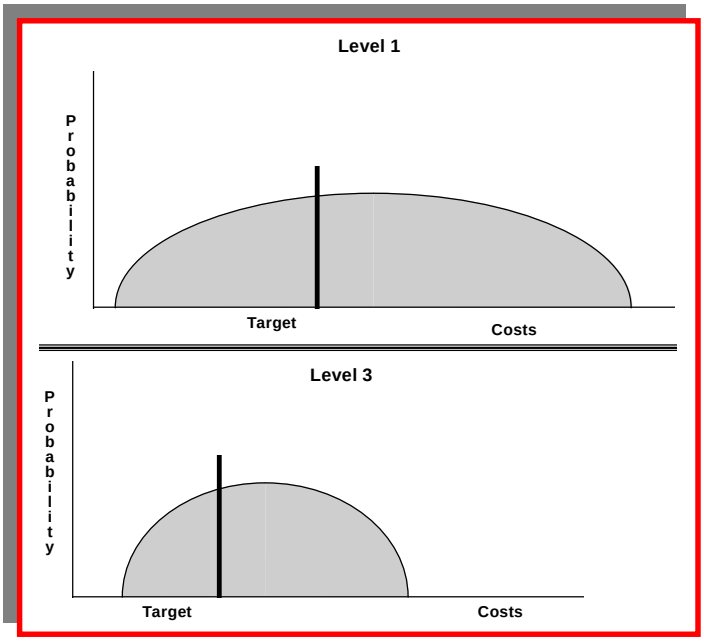

Figure S-3

As shown in Figure S-3, the SEI Level 1 offeror is much more likely to be quantumly incorrect in its estimate than the SEI Level 2 offeror. Moreover, the probability is that the SEI Level 1 offeror will underestimate the cost of the project.

Routinely, where the SEI Level 1 vendor overestimates a project, its proposal is at a competitive disadvantage and rarely receives the award. Conversely, where the SEI Level 1 vendor underestimates a project, its proposal gains a competitive advantage and frequently wins the award. The net result is that, over a period of time, the Level 1 vendor's portfolio of contracts predominately consists of fixed price contracts that are experiencing significant overruns. Anecdotal evidence strongly supports that once a Level 1 offeror begins to lose money on a fixed price contract, the likelihood that the software will be completed to the satisfaction of the agency is greatly diminished.

Unlike most SEI Level 1 vendors, SEI Level 3 vendors have invested heavily into process improvements. Consequently, in terms of reduced overhead, the Level 1 offeror enjoys a conspicuous price advantage. In light of this price advantage, vendors with Level 3 and higher rated processes are reluctant to spend their bid and proposal money on fixed price acquisitions unless the evaluation criteria is structured to favorably consider their superior capabilities and processes. Unwittingly, the government sometimes deters highly competent vendors from submitting a proposal by improvidently selecting the contract type or not astutely drafting the evaluation criteria. (See generally, the discussion below which essentially states that selecting a highly competent vendor is one of the two critical success factors to a software development project meeting the cost, schedule and performance requirements.) Plain and simple, discouraging the best qualified offerors to submit proposals greatly decreases the probability of a successful software development effort.

S7.3.3 Performance Risks

Cost risks, performance risks, and schedule risks are generally interdependent notwithstanding that they are usually addressed separately in risk mitigation plans. The interrelationship between cost and performance risks is exemplified by the following observation of the Court of Claims:

> *"[C]ontractors are businessmen, and in the business of bidding on Government contracts they are usually pressed for time and are consciously seeking to underbid a number of competitors. Consequently, they estimate only on those costs which they feel the contract terms will permit the Government to insist upon in the way of performance."*

This observation is particularly applicable to software development contracts. It is a common attribute of the software development process that the preponderance of the requirement must be decomposed before a comprehensive specification can be drafted. Without a thorough specification to uniformly bind all offerors to a common baseline of performance, the competitive pressure to underbid competitors motivates offerors to only bid what the government *can "insist upon by way of performance*." Once the contract is awarded, the tact the vendor took to win the contract usually necessitates that the contractor contest the allocation of performance risks in each instance where the specification is not abundantly clear. Hence, it is not until late in the software development cycle that the government can effectively shift the risk of nonperformance onto the contractor.

An environment where a vendor frequently contests what the government perceives as a contractual obligation is not only disruptive to the smooth progression of work, but also it can be inimical to the much needed congenial relationship between the parties. The following is an extract of a decision of the Armed Services

Board of Contracts Appeal which exemplifies the debilitating bickering that can arise where a fixed price contract is used without a detailed specification:

> *"The lesson-learned is abundantly clear but frequently overlooked. Before the government selects a fixed price contract ostensibly to place the cost risks on a vendor, the government should scrupulously examine the specification to assure that the performance risks are unequivocally passed to the vendor."*

A fixed price contract only encourages a contractor to perform the bare minimum since anything more must be paid from potential profits. The incentive to perform only the bare minimum is especially strong where the contractor begins to lose money on the venture. Accordingly, a fixed-price contract does not necessarily motivate the contractor to make the refinements that exceed the bare minimum. An example would be where the contractor is obligated to prepare the software requirements specification. Any money saved using a fixed-price contract might be a false savings since a substandard specification can be ruinously expensive to correct later in the software development process where, for lack of diligence, previously undiscerned requirements are discovered. Another example would be late in the software development cycle where the software satisfies the functionality of the software requirements specification but the software needs some minor enhancements to be user-friendly. Such enhancements are less likely to be made voluntarily where there is a fixed price contract.

S7.3.4 Changes Can Impact The Previous Allocation Of Performance Risk

An endemic problem with large software development projects has been excessive changes. Generally, the causes of excessive changes are either a substandard requirements analysis or requirements that are too dynamic to be effectively "*frozen*" into a specification. A landmark GAO report documented why changes frequently are not recognized as contributing to performance risk:

- *"Changes requested after projects have started, which seem trivial to the customers, have often required major rework and have resulted in delays and increased costs.*
- *Changes are not usually as thoroughly researched as original design concepts and sometimes have unforeseen effects on other parts of the system.*
- *Effective use of contract phasing can be destroyed by constantly making changes to work that was competed and approved in earlier phases."*

Additionally, because software is generally perceived as pliable, the users frequently do not appreciate the cost impact of seemingly minor changes. The National Research Council has observed:

> *"Late discovery that some required functions intended to be implemented in hardware cannot practically be so implemented and are shifted to software. This shift might not occur save for the prevalent optimistic view of the pliability of software. In truth, software is not pliable in large, complex systems; a small change in software function can ripple through many interfaces amounting to a major redesign effort, particularly if the added function was not anticipated during the decomposition of CPCIs and modules."*

With regard to the cost risks, it is a fundament rule of government contracts that the contractor is entitled to compensation for the "*unanticipated and extra out-of-pocket expenses it incurred in performing the contract*

as a result of the changes." It is not so widely recognized that changes can sometimes impact the previous allocation of performance risks. Specifically, where the government has crafted the contract to place the performance risks on the contractor, changes that require a contractor to perform in a manner different from what the contractor originally intended can transfer the performance risks from the changes to the government.

S7.3.5 Performance Risk Regarding Architecture

Within the past several years, there has been increased recognition that fundamentally flawed architectures are one of the leading causes of fatality among large software development contracts. Where the architecture is fundamentally flawed, the consequences are catastrophic — often the entire project is either abandoned or restarted. A technique to reduce the risk of a flawed architecture is to require offerors to submit a preliminary software architecture in their technical proposals.

The risk of a fundamentally flawed architecture is particularly high for unprecedented projects. In those instances, a proven technique for the government to reduce performance risks is to award parallel development contracts with two different vendors who propose dissimilar architectures. Typically, sometime between the preliminary design review and the critical design review, the agency exercises a "*down-select*" decision to proceed with only one of the two contracts. When the decision is made is usually a factor of how apprehensive the government is about the design, the criticality of the software in terms of the agency's mission, and the availability of money to continue funding two contractors. The down-select decision is normally exercised in the form of an option to the contract of the selectee.

In addition to the obvious advantage of not having to select an architecture until it has been analyzed in detail, the use of parallel development contracts offers another benefit to the government. Experience has shown that when contractors recognize that they are in competition for the privilege of retaining the project for its life cycle, the vendors are significantly more conscientious about the quality of their work. The competing vendors are also more likely to assign their best software engineers to the project. Ironically, despite the substantial advantages to parallel development contacts, they are rarely used within DoD for large software development projects. The most frequent reason for not using parallel development contracts is failure of the agency to budget adequate money. In retrospect, the failure to make the investment in parallel development contracts for the software architecture has often been regretted.

S7.3.6 The More Participatory The Government Is In The Design, The More Difficult It Is to Shift Performance Risks to The Contractor

The general rule is that the party that has responsibility for the design of a system is accountable if the design results in the failure of the system. Not surprisingly, this rule causes the government to have a preference for performance specifications. A problem arises where the government uses a performance specification but insists upon a highly participatory role in the design of the system. This situation can place the contractor in a dilemma because the government can thwart a contractor from proceeding by failing to approve a review. Although the government can use reviews to "*hold the design in hostage*," the government has cleverly defined the term "*approval*" to distance itself from sanctioning the design.

Research has not disclosed any cases involving software development contracts where the government's participation in the design has caused the government to assume some of the responsibility for the performance risk. There are other decisions, however, which establish this principle of law. For example, the NASA

Board of Contracts Appeal rejected an attempt to hold a contractor completely liable for design flaws that hampered the construction of a scientific facility. Specifically, the NASA Board stated:

> *"In our opinion, this theory completely ignores the elaborate Government organizational structure for both design and construction of facilities, the review and approval requirements built into the contract, the pervasive role of the JSC project engineer who also served as the Contracting Officer's representative.'*

Similarly, the Armed Services Board of Contracts Appeal was unwilling to hold Boeing responsible for the cost of redesigning a fuel-drainage system for the KC-135A aircraft where the government expressed safety concerns after the critical design review.

> *"[T]he Government was anything but passive in monitoring and approving appellants Preliminary Design as it pertained to the drainage. * * * The highly structured dialogue between the Government and appellant generated by the Critical Design Review defined the more detailed Part II Development Specification."*

In light of the above precedent, when planning its acquisition strategy, the government should first ascertain if it intends to play a pro-active role in the design before the government agrees to pay a premium ostensibly to place the performance risk on the contractor. For example, the government should first ascertain if it intends to rely heavily on a Federally Funded Research and Development Center (FFRDC) as an assertive systems engineer or if it intends to rely heavily on a pervasive IV&V contractor. In essence, the highly participatory role of a FFRDC or an IV&V may impede the government from placing the performance risks on the contractor. Stated differently, if the government wishes to shift the performance risk to the contractor, the contracting officer should assure that the government's technical representatives are merely reviewers of the contractor's work rather than participating in the design of the software.

S7.3.7 Schedule Risks

According to Doctor Fredrick Brooks, "*more software projects have gone awry for lack of calendar time than for all other causes combined*." Before probing into the reasons why schedules often doom software development projects, it is important to appreciate the interrelationship between cost, performance and schedule risks. With regard to schedule impacting costs, there is a clear correlation between the number of people on a project and their productivity. In essence, there are efficiencies to be gained when a dedicated but small workforce methodically develop software over a lengthy period of time. The schedule can be expedited, to some degree, by adding additional software engineers. The following table of a hypothetical project is indicative of the dependent relationship between cost and schedule:

SCHEDULE (Months)	SOFTWARE ENGINEERS	STAFF MONTHS	COSTS
9	30	270	$4,500,000
12	20	240	$4,500,000
15	14	210	$4,000,000
18	10	180	$3,750,000

Table S-1

The correlation between schedule and performance risks is that, in attempting to meet an unrealistic schedule, contractors often expedite the process in a matter that is harmful to quality. Frequently, the resulting product is too defective to perform as required. That GAO has repeatedly observed that unrealistic schedules increase performance risk:

> *"Technical problems result from the need to meet deadlines — programs are often designed and written hastily, and are tested and documented inadequately or not at all. Thus quality is sacrificed to urgency. Documentation — material prepared to explain a computer program — is often deferred until after the program is running and sometimes is never completed. When programs are later modified or converted, the work is usually done by someone other than the originator. If documentation is missing, incomplete, or obsolete, a great deal of the original development work often must be repeated."*

As shown above, the schedule can force quality to be sacrificed for urgency. When quality is sacrificed, often the software is degraded or rendered unusable.

In the past, many DoD software projects succumbed to unrealistic schedule that were generated under the euphemism of being "*success oriented*." In 1994, the Air Force published an excellent handbook which acknowledged that unrealistic schedules had a debilitating effect on software development projects. The handbook explains why "*success-oriented schedules are seldom successfully achieved*." The handbook also provides some useful guidance on what it calls "*schedule-plus contracts*." In essence, these contracts are structured to use award fees or incentive fees to motivate a contractor to be realistic in bidding schedules. Equally as important, "*schedule-plus contracts*" are not as likely to cause the contractor to "*sacrifice quality to urgency*."

S7.3.8 The Evaluation Criteria Should Be Structured to Maximize the Probability of Selecting a Highly Competent Vendor

One expert has observed that "the competency of the contractor is the single most important ingredient in the recipe for successful contract performance." For software development contracts, it is axiomatic that the greater the competence of the software development contractor, the greater the probability that the software development project will be successful. Consequently, it behooves the government to structure the evaluation criteria to maximize the probability of selecting a highly competent vendor.

In his book, The Decline And Fall Of The American Programmer, Edward Yourdon summarizes the startling results of some careful studies which reveal that there can be an enormous variation between the capabilities of software engineers. Equally as surprising, there is no simple means to readily distinguish between the top quartile and the bottom quartile of software engineers:

> *"When a programmer is good,*
> *He is very, very good,*
> *But when he is bad,*
> *He is horrid."*

This conclusion was based on the results of a programming exercise given to a group of 12 experienced programmers. Careful records were kept to see how long the programmers took to finish various phases of the programming job, and what results they produced. The outcome was staggering: the best person in the

group finished coding and debugging the exercise 28 times faster than the worst person, and the best program was approximately 10 times more efficient (in terms of memory and CPU cycles) that the worst. Equally important was the discovery that the actual performance of the programmers had no significant correlation with years of programming experience or scores on standard programming aptitude tests.

In the same way that there are an enormous variations among the capabilities of software engineers, so too there are enormous variations among the capabilities of software development vendors. Moreover, just as there is no simple way to readily discern which software engineers are in the top quartile, so too there is no simple way to readily discern which software development vendors are in the top quartile. The following discussion is to provide guidance on how the government can distinguish the relative competence of software development vendors.

S7.4 Software Engineering Institute's Software Capability Evaluations

The SEI's Software Capability Evaluations (SCEs) enable a contracting activity to appraise the maturity level of the offerors. Since ample guidance can be obtained from the SEI on SCEs, this paper will not explain the intricacies of how SCEs are conducted. Suffice it to say that SCEs are expensive and burdensome for both the offerors and the government. As a rule of thumb, a SCE is appropriate where the cost of the software development is expected to exceed ten million dollars or where more than 50,000 lines of code are expected. An exception to the rule might be appropriate where there is a critical need for the software or where human life would be in jeopardy if the software failed.

There are various ways in which a SCE could be considered by the source selection authority. One technique is to make the SCE an affirmative responsibility criteria. For example, the evaluation criteria could state: "*To be eligible for award, an offeror must attain, through a Software Capability Evaluation, Maturity Level 3 or higher*." Before using such an evaluation criteria, the procuring activity should recognize that a high standard such as Level 3 might be challenged by a protest. Under the Competition In Contracting Act, agencies must *achieve "full and open competition*." The Federal Acquisition Regulation defines this term to mean *that "all responsible sources are permitted to compete*." To have the protest denied, the agency must show that its minimum requirement is for a Level 3 offeror. Since Level 3 represents less than 10% of the vendors in the software development industry, the agency should have a convincing reason why a Level 2 vendor is not a responsible source. Rather than take the risk that the GAO or GSBCA would disagree with the agency's justification to exclude lower level offerors, contracting activities should consider other approaches which are easier to reconcile with the Competition in Contracting Act.

Another technique is to establish the rating which the offeror obtains during the SCE as a separate evaluation criteria. Hence, using the above scenario, a Level 2 offeror is not automatically excluded form consideration. Instead, with regard to that evaluation factor, a Level 2 offeror is placed at a competitive disadvantage in relationship to a Level 3 offeror. To have a significant impact on the source selection decision, the evaluation criteria involving the SCEs should be placed relatively high in the relative order of importance among the other evaluation criteria.

A third possibility is to make the SCE a "*general consideration*." A general consideration is factor that permeates the other evaluation criteria. The most widely used general consideration is past performance. There is a logical correlation between an SCE and past performance. Each is a reliable indicator of future success. This logical correlation evinces that an SCE is highly suitable to be a general consideration. The

disadvantage of using an SCE as a general consideration is that it does not afford the SCE the dignity it deserves in playing a pivotal role in the selection of the awardee. Stated differently, a SCE deserves more visibility since the maturity of a vendor's processes is perhaps the paramount indicator of whether the vendor will be able to successfully develop the software. Moreover, if a source selection authority inadvertently relies directly on the SCE in making a selection rather than factoring the SCE into his assessment of the established evaluation criteria, a disappointed offeror might be able to protest successfully.

As stated in the prior paragraphs, there are various ways in which an agency can consider an SCE in making its source selection decision. The method that the agency intends to use should be identified in Section M of the RFP. The Comptroller General will sustain a protest where the offerors have not been advised of the relative order of importance of the evaluation criteria:

> *"It is fundamental that offerors must be advised of the basis upon which their proposals will be evaluated. A solicitation that does not set forth a common basis for evaluating offers, which ensures that all firms are on notice of the factors for award and can compete on an equal basis, is materially defective."*

In summary, it is imperative that Section M of the solicitation clearly state how the agency will use the SCE as part of the evaluation process.

S7.4.1 Past Performance

A vendor who has had a consistent history of successfully performing software development efforts is more likely to successfully perform on future software development efforts than a rival vendor who has had a checked history. Unlike a SCE, which only validates that a vendor has the necessary processes, past performance is a strong indicator of whether the vendor has the fortitude to "*make it happen*."

In recent years, the federal government has become more adamant about relying on past performance in awarding contracts. In a policy letter dated January 11, 1993, the Office of Federal Procurement Policy (OFPP) mandated that past performance be an evaluation factor for all competitively negotiated contracts that were expected to exceed $100,000. The letter also directed the larger agencies to create databases on vendors' past performance. Additionally, in 1994, the OFPP Director obtained pledges from fourteen agencies that they would weigh past performance equally with the other nonmonetary evaluation criteria.

Agencies are afforded considerable discretion in making judgments on past performance. For example, an Air Force procurement of training devices for the F-15 and F-16 aircraft rated an offeror as high risk. The offeror had received more poor performance evaluations than favorable evaluations on previous contracts. The offeror protested to the GAO. In denying the protest, the GAO concluded that the Air Force's risk assessment was reasonable. Additionally, sometimes an offeror seeks to contest a poor rating that it received on another contract. A disappointed offeror has little recourse when it wishes to dispute an unfavorable evaluation. The fact that a disappointed offeror disagrees with an agency's evaluation of its past performance does not invalidate the agency's conclusion.

S7.4.2 Previous Experience

Like past performance, previous experience is a credible indicator of the likelihood that an offeror can successfully perform. For instance, if a Level 1 vendor struggled to complete a previous software project, that vendor still might be preferable to other Level 1 offerors for a comparable project merely because the vendor probably learned many lessons which will benefit it on a subsequent project.

Ideally, the government desires a vendor with both previous experience and an excellent record of past performance. The source selection official faces a more difficult decision where the first offeror has had comparable previous experience but also has had a checked record of past performance. A second offeror lacks previous experience in the domain involving the specification but does have a laudatory past performance record. There is no textbook answer as to which offeror should be selected. The government should, however, anticipate such a quandary and draft its evaluation criteria to accurate forewarn offerors which evaluation criteria is more important.

Previous experience is sometimes expressed as a definitive responsibility criteria. Although definitive responsibility criteria are apt to be protested by vendors who are excluded from competing, the GAO will uphold the agency's decision if it is reasonable. For example, the GAO upheld the decision of the Air Force to exclude an offeror who lacked personnel that were experienced in maintaining a land mobile radio system. The following extract from the decision reveals the GAO's willingness to defer to the agency on definitive responsibility criteria involving previous experience:

> *"Given the agency's explanation . . . that the equipment involved here is used by those units at the base responsible for human safety and the safeguarding of information relating to national security, we have no basis for objecting to the imposition of the experience requirements. Specifically, we see nothing improper in the agency's taking steps to insure that the personnel maintaining the specialized equipment are particularly well-qualified to do so, and the experience requirements seem to us to be reasonably aimed at achieving this end."*

Although the GAO is deferential to an agency's determination of the requisite experience an offeror should have, the GAO will sustain a protest if the agency's position is unreasonable. For instance, in acquiring software support for a shipboard command and control system, the Navy required "*detailed knowledge of the JOTS II Plus program*." An offeror who had a detailed knowledge of an earlier version of the software was excluded from the competition. The GAO sustained the protest of the excluded offeror. Sometimes an agency errs on the side of caution when establishing its minimum experience requirements. The agency later determines that, although a proposal that does not comply fully with the solicitation, the proposal is capable of meeting the agency's requirements. If the agency seeks to award to an offeror who does not meet the minimum experience requirements of the solicitation, the award can be successfully protested. In those circumstances, the correct action for the agency is to reduce the minimum experience requirements by amending the solicitation and then seeking a new BAFO.

S7.5 Sample Problems

According to Gabig's Premise, "*best brochuremanship — not best value — frequently wins government contracts*." Not surprisingly, many sophisticated vendors use professional proposal writers to respond to RFPs. The net result is that the quality of the contractor's proposal may not be indicative of the quality of the technical staffs that will ultimately perform the work. A useful technique to minimize the impact of

Gabig's Premise is to use sample problems. One possibility is to identify as sample problems a few modules that are suitable candidates for rapid prototyping. Another possibility would be to draft the sample problem to address some difficult interfaces that are anticipated during the performance of the contract.

It is not uncommon for a disappointed offeror to protest that its poor score on the sample problems should have been brought to its attention and that it should have given an opportunity to revise its answer. To protect itself again such protests, the agency should draft the solicitation to emphasize that the purpose of the sample problem is to test the offeror's understanding of the problem as well to test the offeror's technical competency. Under those circumstances, as shown by the following quote from a GAO decision, the protest is likely to be denied:

> *"It is also apparent that the Air Force wanted to gauge the offerors' independent management and technical abilities and expertise to propose, on their own, solutions to a variety of complex engineering tasks. While the pointing out of deficiencies in the proposed solutions might well have produced improvements in the offerors' approaches, what was to be evaluated here was not how well an offeror could improve the problem areas, but rather how well an offeror could independently size up a problem and come up with a viable, efficient solution."*

Consistent with the theme that sample problems are a test, they should be given letter grades — A,B,C,D, or F. A pass/fail grading scheme is vulnerable to being struck down as being inconsistent with the nature of negotiated procurements.

S7.6 Avoiding "*Buying-In*"

FAR § 3.501-1 defines buying-in as submitting an offer below anticipated costs, expecting to: (a) increase the contract amount after award (e.g., through unnecessary or excessively priced change orders); or (b) receive follow-on contracts at artificially high prices to recover losses incurred on the buy-in contract. Software development contracts are especially vulnerable to "*buying-in*." In addition to having a high incidence of changes, the opportunity for follow-on maintenance contracts at artificially high prices is particularly great.

Many vendors are unaware that the FAR does not prohibit "*buy-in*." Instead, the FAR only admonishes the contracting officer *to "take appropriate action to ensure buying-in losses are not recovered by the contractor*." For a cost-reimbursement contract, the best way to protect against buy-in is to use a vigorous cost realism analysis. The FAR recognizes that for *"cost-reimbursement contracts the cost proposal should not be controlling, since advance estimates of cost may not be valid indicators of final costs.*" Consequently, an agency is granted considerable leeway to reach an independent evaluation of what it will cost the offeror to complete the project.

An excellent example of an agency asserting its prerogative to adjust a cost proposal for cost realism occurred during an Air Force procurement for software development to support the Joint Space Intelligence Center in Cheyenne Mountain. The Air Force made a $29 million cost realism adjustment to a $69.7 million proposal from McDonnell Douglas Electronics Systems Company. A subsequent protest by McDonnell Douglas was denied by the GAO.

If the contract is fixed price, the best way for an agency to protect against a buy-in is to use a best value procurement. A best value procurement provides the agency with greater flexibility to make tradeoff

decisions. Moreover, the cases involving best value have upheld an agency's decision to spend considerably higher sums of money to achieve superior technical performance.

For a complex software development project, it is not enough that the contracting officer merely comply with his obligation under the FAR to guard against the contractor being able to recover its losses. Buying-in usually results in an antagonistic relationship between the parties since the contractor typically is looking for a way to "*get well*." Conversely, the government obligated to resist the contractor's attempts to "*get well*." In light of the fact that a congenial relationship is almost indispensable to the successful completion of the project, it is in the government's interest to avoid awarding to an offeror who is buying-in if it is at all possible.

Notwithstanding the techniques mentioned above, often the evaluation criteria greatly constrain a source selection official's flexibility to avoid awarding the contract to an offeror who is buying-in. Although rarely explained to the source selection official, there are two other factors which may be considered despite the factors not being expressed in Section M. The first factor is that an agency may infer that the risk of poor performance increases where a contractor is forced to perform a contract with little or no profit. This observation is consistent with the FAR. FAR § 15.901(b) recognizes that profits "*stimulate efficient contract performance*." Inferentially, a lack of profit suggests that the contractor is not stimulated to perform the contract efficiently. The second factor is to assume that the contractor will be forced to pay its workforce low compensation. The source selection official may assume that the low compensation will impacting the offeror's ability to recruit and retain highly quality software engineers. The anticipated difficulty for the offeror to recruit and retain a high quality workforce is justification to increase the agency's assessment of the risk of nonperformance.

S7.7 About the Author

Jerome Gabig, Jr. is Of Counsel in the Washington, D.C. office of Venable, Baetjer, Howard & Civiletti where he specializes in government software development contracts. Through a variety of assignments as an Air Force officer, he gained considerable experience in contracting for software development:

1990-92	Deputy Staff Judge Advocate, Electronic Systems Center (Air Force Materiel Command), Hanscom AFB, MA
1990-87	Director of Contract Law, Armament Division (Air Force Systems Command), Eglin AFB, FL
1987-83	Staff Judge Advocate, Air Force Computer Acquisition Center (Air Force Communications Command), Hanscom AFB, MA
1983-80	Director of Telecommunications & Acquisition Law, Air Force Communications Command, Scott AFB, IL

Mr. Gabig is the course director for Federal Publications' program on software development contracts. Additionally, he is the author of Federal Publications' text entitled Government Contracting For Software Development. Mr. Gabig has graduated from West Point (engineering), Harvard University (management & administration), and the University of California (law). He has instructed at George Washington University, the Army JAG School, the Air Force JAG School, the Defense Systems Management College, the DoD Computer Institute, and the Naval Post Graduate Institute. He performed extensive research and analysis to support the Section 800 Panel. Mr. Gabig founded the Information Systems Committee of the American Bar Association's Public Contract Law Section which he currently serves as Vice-Chair. He is a National Contract Management Fellow and recipient of the Delaney Award for 1993. His numerous publications have appeared in The Harvard Journal of Law & Public Policy, The Public Contract Law Journal, The National Contract Management Journal, Program Manager, Contract Management, and The Computer Lawyer.

Jerome S. Gabig, Jr.
Venable, Baetjer, Howard & Civiletti
Suite 1000
1201 New York Avenue, N.W.
Washington, D.C. 20005
Voice: 202-962-4953
Fax: 202-962-8300
Internet: JGabig@Venable.com